TEKNILLINEN KORKEAKOULU

Sähkötekniikan osasto

Ahti Rissanen

ÄLYVERKON KYTKENTÄPISTEEN SIMULOINTITYÖKALUN SUUNNITTELU JA TOTEUTUS

Diplomityö, joka on jätetty tarkastettavaksi diplomi-insinöörin tutkintoa varten Espoossa 12.09.1994.

Työn valvoja Professori Leo Ojala

TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ

Tekijä: Ahti Rissanen

Työn nimi: Älyverkon kytkentäpisteen simulointityökalun suunnittelu ja toteutus

Päivämäärä: 12.09.1994 Sivumäärä: x + 53

Osasto: Sähkötekniikan osasto

Professuuri: Tik-79 Digitaalitekniikka

Työn valvoja: Professori Leo Ojala

Työn ohjaaja: DI Tapani Karttunen

Diplomityön tarkoituksena on toteuttaa osa kansainvälisen standardisoimisjärjestön suosituksesta älyverkkoarkkitehtuuriksi palvelun kytkentäpisteessä. Toteutetussa ratkaisussa käytetään lisäksi osia Nokia Telecommunications Oy:n toteuttamasta älyverkon sovellusprotokollasta.

Työssä käydään ensiksi lävitse älyverkkojen periaatteellinen rakenne, sen jälkeen selvitetään simulointityökalun toteutuksessa käytetyn ohjelmiston ominaisuudet. Toteutettavan suosituksen sisältö ja toiminnot esitetään toteutuksen eri tasoilla. Simulointityökalun toteutus esitetään yksityiskohtaisesti. Lopuksi pohditaan simulointityökalun jatkokehityksen pääsuuntia.

AVAINSANAT: Älyverkko, kytkentäpiste, simulointityökalu, IN, SSP

HELSINKI UNIVERSITY OF TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author: Ahti Rissanen

Name of the thesis: Designing and implementting simulation tool for Intelligent Networks Service Switching Point

Date: 12 September, 1994 Number of pages: x + 53

Faculty: Electrical Engineering

Professorship: Tik-79 Digital Systems

Supervisor: Professor Leo Ojala

Instructor: M.Sc. Tapani Karttunen

The aim of this thesis is to implement a part international recommendation of Intelligent Networks for service swithing point. Also parts of the Finnish national Intelligent Network Architecture implemented by Nokia Telecommunications Oy have been used.

As an introduction to Intelligent Network basic architectual concept is outlined, then the software tool used to implement service switching point is introduced. The content and functions of the recommendation to be implemented are described in all the various levels of recommendation. Simulation tools implementation is then described.

Finally further development of the simulation tool is outlined.

KEYWORDS: Intelligent Networks, service switching point, simulation tool, SSP, IN

ALKULAUSE

Tämän diplomityön valvojana on toiminut professori Leo Ojala, jolle haluan esittää parhaimmat kiitokseni. Hänen kanssaan käymäni keskustelut ovat suurelta osin vaikuttaneet diplomityön sisältöön.

Työn ohjaajana toiminutta DI Tapani Karttusta haluan kiittää työn eteenpäinsaattamisesta. Hänen ehdotuksensa työn priorisoinnista ovat selkeyttäneet työn rakennetta.

Lisäksi haluan esittää kiitokseni Telecom Finland Oy:n Telekehityskeskuksen Palvelusovellukset-yksikössä työskenneille henkilöille, joiden kanssa käymäni keskustelut ovat omalta osaltaan vaikuttaneet työn loppuunsaattamiseen.

Omalta osaltaan työn "edistymiseen" on vaikuttanut myös vaimoni Nina, jota saan kiittää niistä kerroista, joina hän hoputti minua tekemään diplomityötä sekä erityisesti pientä tytärtämme Annia, joka sinnikkäästi jaksoi viedä kirjoittamiseen tarvittavaa tehollista työskentelyaikaa. Lisäksi täytyy mainita koiramme Konrad "Koukku", joka innokkaasti veti minut kirjastoon tiedonlähteille. Maininnan ansaitsevat myös oikolukijat Outi Järvineva, Anneli Rämö ja Pekka Turpeinen.

Espoossa, syyskuun 12. päivänä 1994.

Ahti Rissanen

SISÄLLYSLUETTELO

LYHENNELUETTELO

AD **Adjunct**
Älyverkkokeskukseen liitettävä älykäs lisälaite

AIN **Advanced Intelligent Network**
Älyverkko (Bellcore)

ASN **Abstract Syntax Notation**
Kuvauskieli

B-ISDN **Broadband Integrated Services Digital Network**
Laajakaistainen digitaalinen monipalveluverkko

BCP **Basic Call Process**
Puhelun lohko

BCM **Basic Call Model**
Puhelumalli

BCSM **Basic Call State Model**
Puhelun tilamalli

Bellcore **Bell Communication Research**
Amerikkalainen standardisoimisjärjestö

CC#6 **Common Channel Signalling Number 6**
Eurooppalainen yhteiskanavamerkinantojärjestelmä

CCAF **Call Control Agent Function**
Peruspuhelun ohjausfunktio

CCF **Call Control Function**
Älyverkon puhelunohjausfunktio

CCIS **Common Channel Interoffice Signalling**
Amerikkalainen yhteiskanavamerkinantojärjestelmä

CID **Call Instance Data**
Yksittäisen puhelun parametrit

CIDFP **Call Instance Data Field Pointer**
Puhelun parametrien kenttäosoitin

CS **Capability Set**
Toiminnallinen joukko, ITU-T:n älyverkkosuositusten sukupolvi

CS-1 **Capability Set 1**
ITU-T:n älyverkkosuositusten ensimmäinen sukupolvi

DP **Detection Point**
Havaitsemispiste

DFP **Distributed Functional Plane**
Hajautettu toiminnallinen taso

EFSA **Extended Finite State Automaton**
Laajennettu äärellinen automaatti

ETSI **European Telecommunication Standardization Institute**
Eurooppalainen telealan standardisoimisjärjestö

FE **Functional Entity**
Toiminnallinen olio, SIB:n rakennusosa

FEA **Functional Entity Action**
Toiminnallisen olion lohko, joka ohjaa FE:n toimintaa

FINAP **Finnish Intelligent Network Application Protocol**
Suomalainen älyverkon sovellusprotokolla, jonka Nokia Telecommunications Oy on toteuttanut

GSL **Global Service Logic**
Yleinen palvelulogiikka, määrittelee palvelun yleisen kulun

IN **Intelligent Network**
Älyverkko (ITU-T)

INAP **Intelligent Network Application Protocol**
Älyverkon sovellusprotokolla

INCM **Intelligent Network Conceptual Model**
ITU-T:n älyverkon käsitteellinen malli

IP **Intelligent Peripheral**
Älyverkkokeskukseen liitettävä päätelaite, jota käytetään ilmoitusten toistoon ja äänitaajuusvalintojen vastaanottamiseen

ISDN **Integrated Services Digital Network**
Digitaalinen monipalveluverkko, ITU-T:n suositus puhelinverkkotekniikaksi

ISO **International Standardization Organization**
Kansainvälinen standardisoimisjärjestö

ITU **International Telecommunication Union**
YK:n alainen kansainvälinen teleliitto

ITU-T **Telecommunication Standardization Sector of ITU**
ITU:n teleliikenteen standardisoimissektori

N-ISDN **Narrowband Integrated Services Digital Network**
Kapeakaistainen digitaalinen monipalveluverkko

NMT **Nordic Mobile Telephone**
Pohjoismainen käsipuhelinverkko

OS **Operating System**
Käyttöjärjestelmä

OSI **Open Systems Interconnection**
ISO:n malli avoimeen tietoliikenteeseen

PDU **Protocol Data Unit**
Protokollan tietoyksikkö

PE **Physical Entity**
Fyysinen olio

PIC **Point In Call**
Havaittava piste puhelussa

POI **Point of Initiation**
Palvelulogiikan alkupiste

POR **Point of Return**
Palvelulogiikan paluupiste

PSPDN **Packet Switched Public Data Network**
Pakettivälitteinen tietoverkko

PSTN **Public Switched Telephone Network**
Yleinen puhelinverkko

RBOC **Regional Bell Operating Company**
Amerikkalaisesta alueellisesta teleoperaattorista käytetty nimike

SCEF **Service Creation Environment Function**
Älyverkkopalveluiden luonti- ja testausfunktio

SCEP **Service Creation Environment Point**
Älyverkkopalveluiden luonti- ja testauspiste

SCF **Service Control Function**
Älyverkkopalveluiden ohjausfunktio

SCP **Service Control Point**
Älyverkkopalveluiden ohjauspiste

SDF **Service Data Function**
Älyverkkopalvelun tietofunktio

SDL **Service Description Language**
Palveluiden määrittelyiden kuvauskieli

SF **Service Feature**
Palvelutoiminto

SIB **Service Independent building Block**
Palvelun rakennusosa

SLP **Service Logic Program**
Palvelulogiikkaohjelma

SMF **Service Management Function**
Palvelun hallintafunktio

SMP **Service Management Point**
Palvelun hallintapiste

SN **Service Node**
Älyverkkokeskuksen lähellä sijaitseva palvelupiste

SPC **Stored Program Control**
Tietokoneohjattu puhelinkeskus

SRF **Service Resource Function**
Älyverkkopalveluiden resurssifunktio

SSD **Service Support Data**
Palvelun parametrit

SSF **Service Switching Function**
Älyverkkopalveluiden kytkentäfunktio

SSP **Service Switching Point**
Älyverkkopalveluiden kytkentäpiste

STP **Signal Transfer Point**
Merkinannon siirtopiste

TC **Transaction Capability**
Yhteystapahtumatoiminnot

VPN **Virtual Private Network**
Yleisen puhelinverkon sisällä oleva näennäinen puhelinverkko

1 JOHDANTO

Puhelinliikenteen jatkuva kasvaminen sekä käyttäjien vaatimat monimutkaiset palvelut ovat johtaneet tarpeeseen kehittää menetelmiä ja tapoja, jotka mahdollistavat puhelimen käyttäjälle monipuolisemman tavan hyödyntää hallussaan olevaa liittymää. Puhelinverkkojen digitalisoituminen mahdollistaa verkkojen hyödyntämisen aiempaa laajemmassa mittakaavassa. Digitalisoinnin ansiosta puhelimen käyttäjä on nyt saavutettavissa aikaisempaa helpommin, paremmin ja varmemmin. Tietokoneiden yleistyminen puhelinkeskusten ohjauksessa saa aikaan sen, että olemassa olevia järjestelmiä hyväksikäyttäen voidaan luoda koko maailman kattavia informaatiojärjestelmiä. Nämä järjestelmät ovat jokaisen käyttäjän saavutettavissa ilman, että hänen täytyy sijoittaa uusiin kalliisiin laitteisiin. Yksinkertaisimmillaan puhelimen käyttäjän hallussa jo oleva laite on riittävä edellytys palveluiden hyväksikäyttämiselle.

Älyverkon standardisoimisen tärkeys onkin noussut esille kehitettäessä käyttökelpoisia ratkaisuja puhelinverkkoihin, joissa yleensä on usean laitetoimittajan valmistamia osia. Eri standardisoimisjärjestöt ovat painottaneet erilaisia katsantokantoja älyverkkoarkkitehtuuriin. Vaikka työtä standardisoimisen eteen on tehty jo usean vuoden ajan, vasta nyt on löytymässä käyttökelpoisia ratkaisuja. Syitä eri standardisoimisjärjestöjen kehittämille ratkaisumalleille on useita. Sekä Bellcore (Bell Communication Research), että ITU-T (International Telecommunication Union Telecommunication Sector) tarkastelevat ongelmaa erilaisista näkökulmista. Lisäksi suurimmat laitevalmistajat ovat pyrkineet vaikuttamaan standardisoimiseen omalta kannaltaan edullisimmalla tavalla.

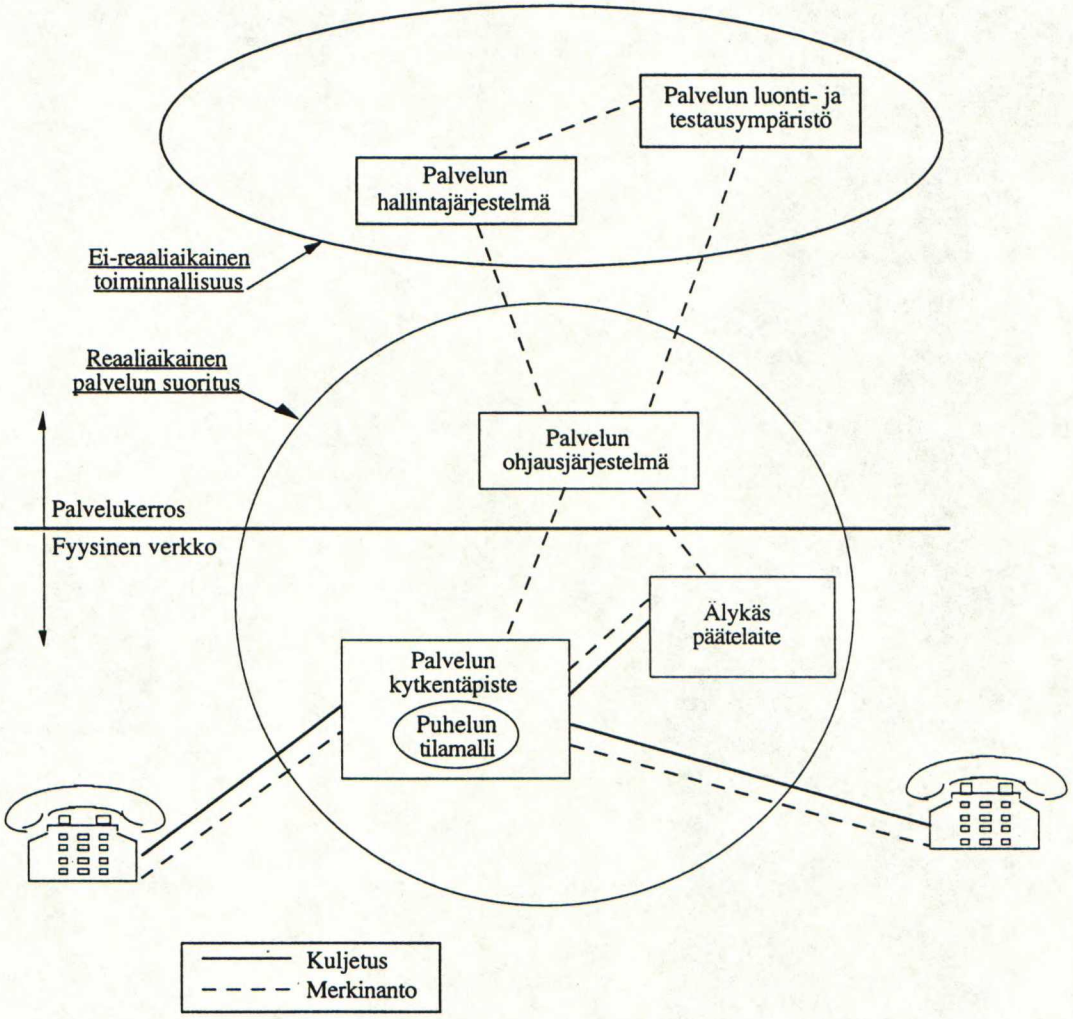
Tässä työssä toteutetaan osa kansainvälisen standardisoimisjärjestön ITU-T:n määrittelemästä suosituksesta älyverkkostandardiksi. Työssä on lisäksi käytetty osia Nokia Telecommunications Oy:n määrittelemästä älyverkon sovellusprotokollasta (Finnish Intelligent Network Application Protocol, FINAP).

Työn alussa kuvataan ensiksi älyverkkojen yleistä rakennetta, sitten esitellään työn tekemisessä apuna käytetty kehitysympäristö. Toteutettavan simulaattorin lähtökohdat esitellään käsitteellisellä ja käytännönläheisellä tavalla. Lopuksi esitetään johtopäätökset sekä suuntaviivat simulaattorin jatkokehitykselle.

2 ÄLYVERKOT

2.1 Yleistä

Termi "älyverkko" (Intelligent Network, IN) otettiin käyttöön yhdysvaltalaisessa teleyrityksessä 1980-luvun puolivälissä. Tähän ajattelutapaan perustuvia palveluita on tosin ollut jo saatavilla 1980-luvun alusta lähtien. Älyverkkojen käsite tarkoitti aluksi keskitetyn älykkyyden lisäämistä yleiseen puhelinverkkoon. Tarkoituksena oli tuottaa ja muokata tiettyjä käyttäjien haluamia palveluita aikaisempaa taloudellisemmin. Nykyään älyverkon käsite on laajentunut tarkoittamaan yleistä arkkitehtuuria, jossa palveluita tuotetaan nopeammin ja edullisemmin kuin aikaisemmin on voitu tehdä. [Li93]



Kuva 1: Yleisnäkymä älyverkkoon [Hom92]

Määritelmä 1: Älyverkko

> "Arkkitehtuuri, joka mahdollistaa lisäarvopalveluiden tukemisen, ohjauksen ja hallinnan ilman, että se olisi riippuvainen käytetyistä teleliikenneverkoista, tarjotuista palveluista sekä verkon laitteista."

Muutamassa vuodessa useat eri tahot ovat tehneet merkittävää työtä määritellessään älyverkkoarkkitehtuuria. Näistä huomattavimmat ovat olleet Bellcore Yhdysvalloissa, ETSI (European Telecommunication Standardization Institute) Euroopassa ja ITU-T kansainvälisesti. Älyverkkojen kehityksen jatkuminen riippuu siitä, miten nämä eri järjestöt yhtenäistävät kehittämiään mallejaan. Tulevaisuuden älyverkkoarkkitehtuurin tulisi perustua näiden järjestöjen yhteisesti standardisoimalle liittymälle. [Li93]

2.2 Historiaa

Älyverkkojen ensimmäiset kehitysvaiheet voidaan jäljittää 1960-luvulle, jolloin teleoperaattorit alkoivat vaihtaa käytössä olleita mekaanisia keskuksia tietokoneohjatuiksi keskuksiksi (Stored Program Control, SPC). Tietokoneita alettiin käyttää puhelinkeskuksien ohjauksessa. Alussa tietokoneohjauksen keskukselle tuoma älykkyys oli keskittynyt lähinnä puhelun kytkentätoimenpiteisiin. [Li93]

Tietokoneistettujen keskusten järjestelmät kehittyivät siten, että 1970-luvun puolivälissä niitä käytettiin myös järjestelmissä, joilla hallittiin ja ylläpidettiin puhelinverkkoa. Yleisesti näitä järjestelmiä kutsuttiin käyttöjärjestelmiksi (Operations Systems, OS). Syynä tietokoneiden sijoittamiselle keskuksiin oli puhelinverkon laajeneminen ja tarjottujen palveluiden määrän kasvaminen. [Ber92]

Teleoperaattorit Pohjois-Amerikassa ja Euroopassa alkoivat samanaikaisesti käyttää yhteiskanavamerkinantojärjestelmää (Common Channel Interoffice Signalling, CCIS, Yhdysvalloissa ja Common Channel Signalling Number 6, CC#6, Euroopassa), joka mahdollistaa älyverkon palveluun liittyvän tiedon erottamisen puhelinverkon keskuksen informaatiosta. Tyypillisesti älyverkon palveluun liittyvä informaatio on tietokannoissa, jotka sijaitsevat puhelinverkon ohjauspisteissä. [Li93]

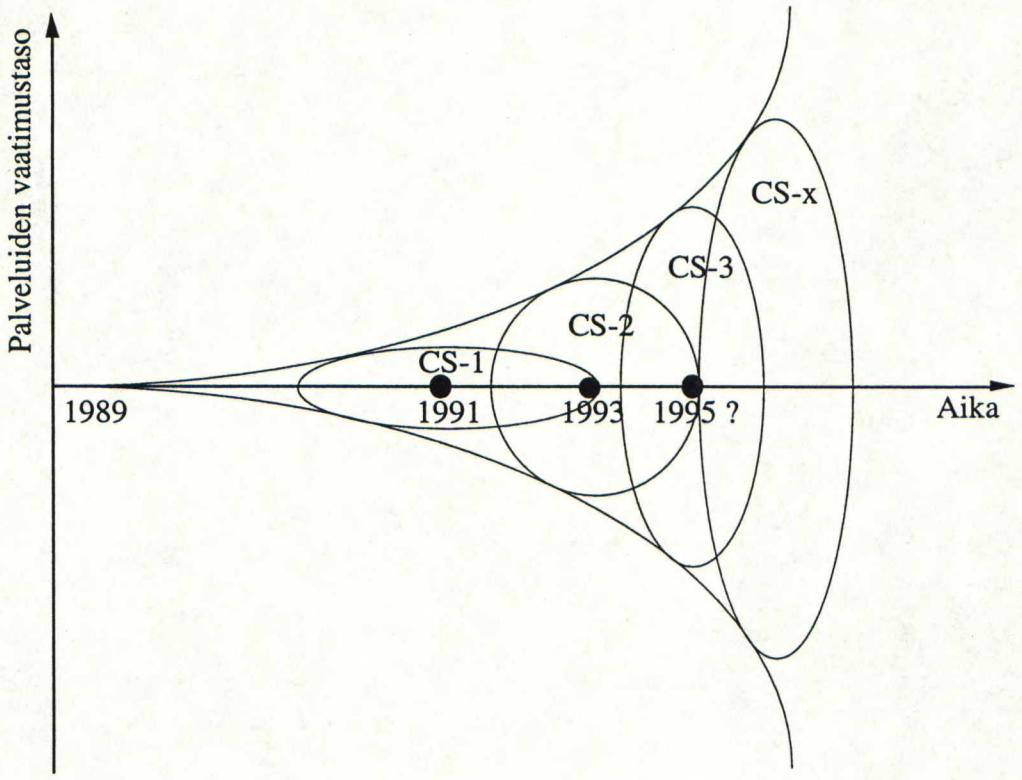
Keskitettyjen tietokantojen käyttöönottaminen mahdollisti sen, että vuonna 1981 eräs amerikkalainen puhelinyhtiö pystyi tarjoamaan 800- eli ns. Freephonepalvelun. Pian myös kilpailevat puhelinyhtiöt esittelivät omat älyverkkopalvelunsa. [Li93]

Vuonna 1984 Yhdysvalloissa aloitettiin työ, jonka tarkoituksena oli erottaa palveluiden toiminteet itse kytkentätapahtumista. Älyverkkojen kehityksen pohjana olleen projektin tavoitteina oli riippumattomuus verkon operaattorista sekä palveluiden nopea ja taloudellinen tuottaminen. [Col92]

Yhdysvalloissa tapahtunut kehitys 1980-luvulla herätti kiinnostusta myös Euroopassa. Vuoteen 1990 mennessä kaikki suurimmat eurooppalaiset teleoperaattorit tarjosivat ainakin 800-palveluita, jotka perustuivat Bellcoren IN/1-arkkitehtuuriin. Tällä hetkellä kiinnostus Euroopassa keskittyy ITU:n hyväksymään suositukseen, joka tunnetaan yleisesti nimellä CS-1. Vaikka ITU-T:n työskentelyyn ottaa osaa teleoperaattoreita ympäri maailmaa, ovat eurooppalaiset, etenkin ETSI:n jäsenet, tehneet merkittävimmän työn älyverkkoarkkitehtuurin standardisoinnissa. [Li93]

Kuten Bellcore, ITU-T:kin kehittää älyverkkoarkkitehtuuria pitkällä tähtäimellä ja useissa eri vaiheissa. Molemmissa standardisoimisjärjestöissä on kuitenkin huomattu, että älyverkkojen kaikkien toiminteiden kehittäminen ja toteuttaminen vie useita vuosia. Lisäksi älyverkkoarkkitehtuurin käsitettä täytyy kehittää siten, että sen pohjana käytetään olemassa olevia laitteistoja ja järjestelmiä. [Li93]

Perusajatuksena ITU-T:n suosituksissa on älyverkon käsitemalli (Intelligent Network Conceptual Model, INCM), joka on kehyksenä älyverkkojen suunnittelulle ja kuvaukselle. Tätä mallia hyväksikäyttäen toteutetaan useita CS-x:n versioita kehityksen eri vaiheita varten. [Li93]



Kuva 2. CS-x:ien kehittyminen [Li93]

2.3 Älyverkkojen periaatteet [Li93]

Kommunikoinnin muuttuessa älyverkot tarjoavat erilaisille televerkon sidosryhmille (operaattoreille, valmistajille ja käyttäjille) uusia mahdollisuuksia hyödyntää keskinäistä viestintäänsä. Jos jokin näistä sidosryhmistä ei hyödynnä älyverkkojen niille tuomaa tilaisuutta, jää se jälkeen kilpailutilanteessa.

Telealan monopoleja purettaessa ja puhelinliittymien lukumäärän kasvun hidastuessa täytyy teleyritysten keskittää markkinointinsa lisäarvopalveluihin. Näiden kehittäminen on sekä puolustuskeino, taistelua kilpailijoita vastaan, että myös kasvualusta, laajentumiskeino uusille markkina-alueille ja markkinoille. Älyverkkoarkkitehtuurin yhtenäisyys mahdollistaa sen, että teleoperaattorit voivat tarjota uusia, kaikkien käyttäjien saatavilla olevia lisäarvopalveluita, jotka kuitenkin on räätälöity asiakkaiden vaatimusten mukaisesti.

Useiden toimittajien valmistaessa tuotteita yhtenäiseen älyverkkoarkkitehtuuriin ei teleyritysten tarvitse sitoutua jonkin tietyn laitetoimittajan tarjoamiin ratkaisuihin. Älyverkkoarkkitehtuuriin siirtyminen ei kuitenkaan varsinaisesti vähennä keskustoimittajien mahdollisuuksia, sillä siirtyminen älyverkkopohjaisiin

keskuksiin vaatii vanhojen kytkentäkeskuksien muuttamista. Näistä muutostöistä avautuu todennäköisesti huomattavat markkinat laitevalmistajille. Myös muut älyverkon osat — palvelun ohjauspisteet, älykkäät laitteet, palvelupisteet, palvelun hallintajärjestelmät — tuovat uusia mahdollisuuksia laitteistojen ja ohjelmistojen toimittajille. Palveluiden tuottajille ja hyväksikäyttäjille älyverkot tuovat kustannussäästöjä, parempaa palvelua sekä paremman mahdollisuuden hallita yrityksen viestintää.

2.4 Älyverkkojen tilanne [Li93]

Älyverkkoarkkitehtuuri on saavuttanut kehityksessä sen vaiheen, että komiteoiden tekemät päätökset kohtaavat markkinoiden vaatimukset. Onkin tärkeätä, että älyverkkoarkkitehtuurin kehittämistä jatketaan. Älyverkkoarkkitehtuurin kehitystä ajavatkin eteenpäin useat erilaiset sidosryhmät.

Standardisoimisjärjestöt

Standardisoimistyön kannalta on tärkeää, että standardien kehittäminen ei viivästy tämänhetkisestä tilanteesta. Älyverkkoarkkitehtuurin laajuus ei saa häiritsevästi viivyttää toteutuksen yksityiskohtien suunnittelua. Standardien täytyy tarjota yksityiskohtaiset ja tarkat ohjeet niiden toteuttamiseksi. Samalla niiden on kuitenkin otettava huomioon vallitsevien markkinoiden asettamat vaatimukset. Molempien standardisoimisjärjestöjen suositusten tulisikin lähestyä toisiaan. Tämän suuntaista kehitystä on havaittavissa suositusten tulevissa versioissa.

Teleoperaattorit

Teleoperaattorit ovat jo huomanneet älyverkkojen tärkeyden toiminnalleen. Ei olekaan sattumaa, että ne ovat olleet aktiivisimpia älyverkkopalveluita toteutettaessa. Teleliikenteen monopolien poistaminen tulee peruuttamattomasti muuttamaan palveluiden tarjoajien mahdollisuuksia toimia nykyisillä markkinoilla. Oletettavissa on, että tämä kehitys kiihtyy hyvinkin nopeassa tahdissa. Teleoperaattoreiden tulisi olla aggressiivisempia palveluiden tarjonnassaan, jos ne haluavat olla mukana koventuvassa kilpailutilanteessa.

Toimittajat

Älyverkkojärjestelmien markkinat kasvavat vuosikymmenen puolivälissä huomattavasti. Laitetoimittajien tulisikin keskittää ponnistelunsa yhteenliittymiin, jotka tehokkaammin voisivat vastata kilpailuun. Uusia alueita, joiden markkinat ovat avautumassa, sijaitsee Aasiassa, Itä-Euroopassa sekä Etelä-Amerikassa.

Käyttäjät

Vuosikymmenen loppupuolella kaikenkokoiset yritykset käyttävät älyverkkojen tarjoamia palveluita päivittäisessä liiketoiminnassaan. Joitakin tärkeimpiä palveluita tarjotaan myös kotitalouksien käyttöön, tosin nämä palvelut ovat luonteeltaan ns. massapalveluita.

Markkinat

Älyverkkojärjestelmät tuovat enemmän kuin 32 % teleoperaattoreiden tuloista vuoteen 2000 mennessä. Jos kannettavien puhelinten osuutta ei oteta huomioon, prosenttiosuus on hieman alle 30 %. Älyverkkopalvelut on huomattavin tekijä teleoperaattoreiden toiminnassa 2000-luvulla. Eri älyverkkopalveluista 800- ja 700-numerot sekä virtuaaliset puhelinverkot (Virtual Private Network, VPN) ovat suurimmat tulonlähteet. Arvioidaankin, että nämä kolme älyverkkopalvelua yhdessä muodostavat vuosisadan lopulla 75 % kaikista ei-liikkuvan puhelinliikenteen tuloista Yhdysvalloissa ja 80 % Euroopassa. Yleisesti ottaen älyverkkojen kehityksen tärkeimmät painopistealueet ovat standardien ja tuotteiden saatavuus, tarjottujen palveluiden määrä sekä itse tarjottujen palveluiden sisältö.

2.5 Älyverkkojen standardisoimisesta [Li93]

Eri järjestöt ovat tehneet viime vuosina merkittävää työtä älyverkkojen standardisoinnissa. Bellcoren Yhdysvalloissa tekemä työ on tosiasiassa kansallinen standardi, jonka useat teleoperaattorit ja laitevalmistajat ovat valinneet omien toteutuksiensa pohjaksi. Kansainvälisen ITU-T:n CS-suositukset ovat Eurooppalaisen ETSI:n määritysten pohjana. Ensimmäisten palveluiden valmistuminen Yhdysvalloissa sekä useiden eurooppalaisten teleyhtiöiden panostukset CS-1:een ovat lisänneet painetta standardisoimistyöhön.

Olemassaolevat standardit

Älyverkoille on olemassa kaksi erilaista standardia: Bellcoren kehittämä AIN (Advanced Intelligent Network) -standardi ja kansainvälisellä tasolla toimivan ITU-T:n suositusehdotus, jota kutsutaan Q.1200-suositussarjaksi. Vaikka Bellcorella ja ITU-T:llä on yhteinen pitkän tähtäimen näkemys älyverkon standardisoinnista, on niiden lähestymistavoissa eroavaisuuksia. Bellcoren kehittämää AIN:n standardisoimista ohjaavat Yhdysvaltain alueellisten puhelinyhtiöiden (Regional Bell Operating Company, RBOC) tavoitteet. Standardissa olevat määrittelyt heijastavatkin näiden yhtiöiden vaatimuksia ja kokemuksia älyverkoista. Bellcoren AIN:n julkaisut keskittyvät täten standardisoimisessa älyverkon yksityiskohtaiselle tasolle. Ne asettavat myös vaatimuksia, joiden tarkoituksena on tukea tiettyjä palveluita.

ITU-T:n suosituksen lähestymistapa lähtee ylhäältä alaspäin. Tämä selittää työskentelyn älyverkkojen käsitteelliselle mallintamiselle, joka on pohjana älyverkkoarkkitehtuurin pitkäaikaisille vaatimuksille. Koska ITU-T:n työskentelyyn ottaa osaa useiden teleoperaattoreiden edustajia kaikkialta maailmasta, täytyy sen tekemien suositusehdotuksien läpäistä useiden maiden asettamat vaatimukset. Määrittelyjen tärkeimmässä kohdassa, älyverkon sovellusprotokollassa INAP:ssa (Intelligent Network Application Protocol), olevat useat erilaiset lähestymistavat ovat monien teleoperaattoreiden ongelmina. ITU-T ei voi suosituksissaan kuin vain tarjota viitekehyksen älyverkkojen toteuttamiselle.

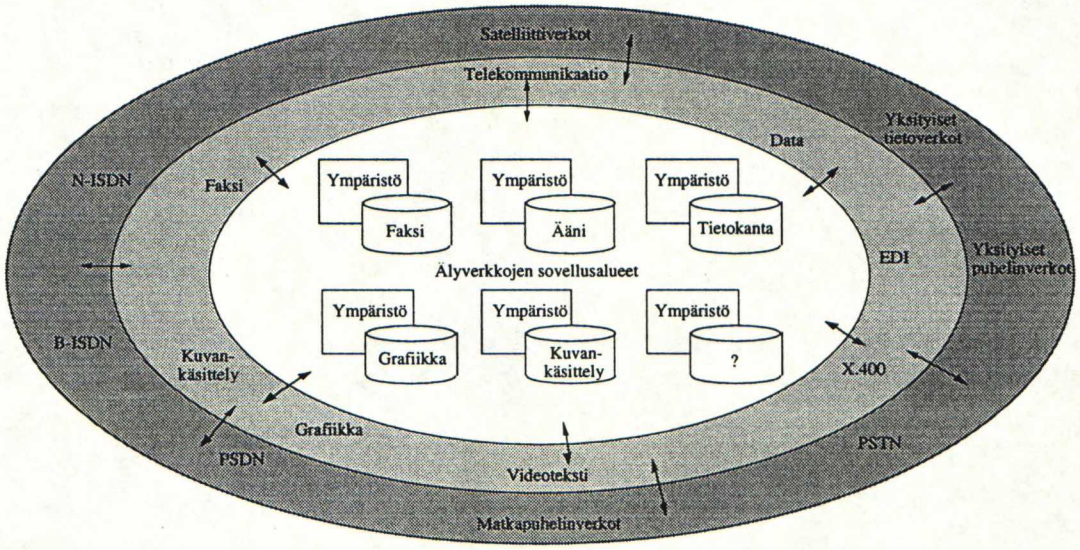
ITU-T:n Q.1200-sarjan suositukset

ITU-T:n Q.1200 sarja jakautuu kahteen luokkaan: yleisnäkymään älyverkoista ja älyverkon käsitteelliseen malliin. Yleisnäkymä älyverkkoihin käsittelee yleiset toiminnalliset vaatimukset, arkkitehtuuriset käsitykset, pitkän tähtäimen arkkitehtuurisen viitekehyksen ja tulevien CS-suositusten määritykset. Älyverkon käsitteellinen malli sisältää itse arkkitehtuuriset vaatimukset, joihin palveluiden tuottaminen perustuu.

Pitkän tähtäimen näkymä älyverkkoihin

Älyverkkojen käsite on rajoittamaton siinä mielessä, että älyverkkoarkkitehtuuri ei aseta sellaisia rajoituksia palveluiden yksityiskohdille, jotka olisivat teknisesti mahdottomia tuottaa. Tulevaisuudessa kaikki telepalvelut voivatkin olla älyverkkopalveluita.

Älyverkkoarkkitehtuuria voidaan käyttää erityyppisten televerkkojen yhdistämisessä ja niiden välisissä liityntäpisteissä. Televerkon osien integrointi voidaan toteuttaa käyttöjärjestelmien elementeillä, joita tukevat älyverkkojärjestelmissä olevien sovellutusten yhteiset infrastruktuurit. Tämän vuoksi ITU-T:n CS-suositukset sisältävätkin televerkon arkkitehtuurin ja sen palveluiden luomiseksi tarvittavien työkalujen kuvaukset. Erilaisia käyttökelpoisia apuvälineitä ovat laajakaistaisuus, hajautettu prosessointi, OSI, objektiorientoitunut mallintaminen, hajautettu ohjaus, televerkon ja sen palveluiden hallinta sekä tekoäly.



Kuva 3: Multimedian yhdistäminen televerkkoon

Pitkän tähtäimen älyverkkoarkkitehtuuri voi myös mahdollistaa multimedian yhdistämisen palveluihin. Tällä hetkellä on tiedonsiirrossa olemassa useita erilaisia standardisoituja liityntöjä ja protokollia, jotka täytyy liittää yhteen yhtenäiseen viitekehykseen. Älyverkot voivat tarjota yhtenäisen ja avoimen alustan, joka tukee tätä viitekehystä.

Älyverkkojen tavoite on samalla sekä kunnianhimoinen että vaativa. Sen toteuttaminen vaatii jatkuvia ponnisteluja ja uhrauksia yksittäisiltä standardisoimisjärjestöiltä sekä jatkuvaa yhteistyötä näiden välillä. On kuitenkin tärkeää, että älyverkon tavoitteita ei karsita toteutusten standardisoimisen vuoksi. ITU-T:ssä tehdyt sinänsä kiitettävät ponnistelut älyverkon käsitteellisen mallin luomiseksi eivät ole riittäviä. Tehtyjen ratkaisujen lisäksi vaaditaan ohjauksen yksityiskohtaista toteutusta ja olemassa olevien arkkitehtuurien muokkaamista. Mitä enemmän valinnaisuutta standardeihin jätetään, sitä vaikeampaa on toteuttajien ja toimittajien työ. Ratkaisujen täytyy olla joustavia, mutta samalla niiden pitää myös

ottaa huomioon käytännön asettamat rajoitukset. Joustava arkkitehtuuri ei välttämättä ole tärkein ehto monen toimittajan ympäristölle.

2.6 Älyverkkojen lähestymistapa 2000-luvulle

Älyverkot ovat tällä hetkellä vasta tulossa markkinoille. Onkin tärkeää, että tämä kehitys saa jatkua keskeytyksettä. Standardisoiminen ei saa viivästyä entisestään. Älyverkkoteknologian hyvyyttä ei saa heikentää standardisoimisessa pitäytymällä liikaa olemassa oleviin toteutukseen. Standardien täytyy antaa yksityiskohtaiset, selkeät toimintalinjat, jotka kuitenkin samalla ottavat huomioon markkinavoimien vaatimukset. [Li93]

Yksi tärkeimmistä CS-1:n tavoitteista on älyverkon muokattavuus. Älyverkkoarkkitehtuurin tulee olla jatkuvasti tarkentuva prosessi. Tämän kehityksen ohjaukseen tarvitaan porrasmainen viitekehys, jonka ohjaamana suunnittelu toteutetaan. Viitekehyksen täytyy sisältää älyverkkojen pitkän tähtäimen tavoitteet. Avainkohtana tällä porrasmaisella tarkastelulla on edellistä tasoa laajempien suositusten määrittäminen. Tarkoitus on, että suositus CS-(x-1) on CS-x:n osajoukko, jolloin kaikki edelliset suositukset eivät jää tarpeettomiksi, vaan ne ovat pohjana kehitykselle. [Dur92] Kun älyverkkojen AIN- ja CS-arkkitehtuurit lähestyvät toisiaan tulee molempien järjestöjen määritellä niiden tarvitsemat resurssit tarkoin. Sekä Bellcore että ITU-T ovat molemmat tunnustaneet tämän tosiasian. Älyverkkojen kehitykselle standardien yhtenäistymisellä on hyvin suuri merkitys. [Li93]

3 ITU-T:n ÄLYVERKKOARKKITEHTUURI

3.1 Yleistä

Standardisoiminen [Li93]

Yhdysvalloissa Bellcore on julkaissut perusarkkitehtuurin, AIN Release 1:n, johon tämänhetkinen laitteistojen ja ohjelmistojen kehitys tähtää. Bellcore on myös julkaissut yksityiskohtaisemmat versiot 0.1 ja 0.2, joiden tarkoituksina on helpottaa siirtymistä versioon 1.0. Tulevaisuudessa on tarkoitus julkaista myös muita versiota 1.0 tukevia välivaiheita.

Myös kansainvälinen telealan yhteistyöelin, ITU-T, on julkaissut ehdotuksensa älyverkkosuositukseksi. ITU-T:n Q.1200-suositussarja sisältää älyverkkojen yleisen osan sekä joukon älyverkkojen määrityksiä, jotka mahdollistavat ensimmäisten palveluiden luomisen. Suosituksessa kuvataan älyverkon arkkitehtuuri.

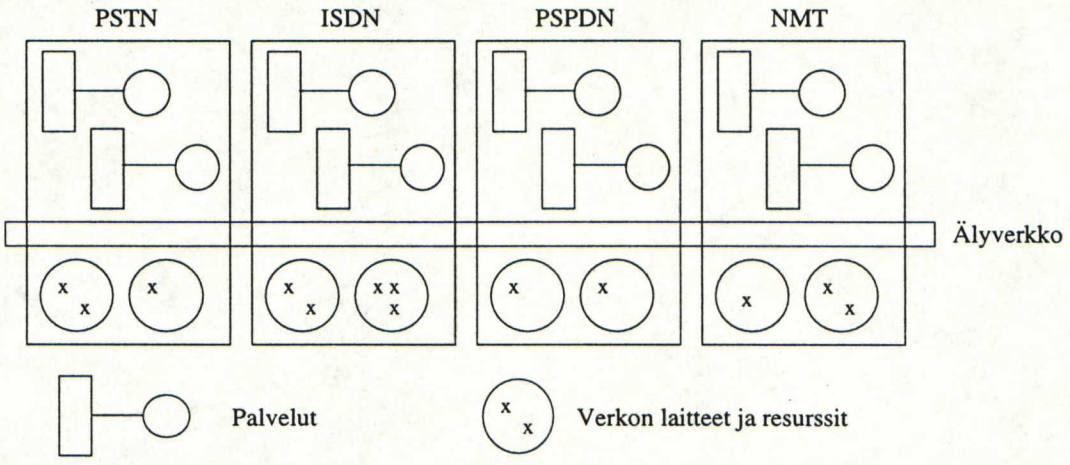
Standardisoiminen kestää tavallisesti kauan, eikä lopputulos yleensä vastaa käyttäjien toivomuksia. Vaikkakin älyverkkojen määrityksissä on edistytty merkittävästi, ei suunnitellussa aikataulussa ole pysytty. Määräajat ovat siirtyneet useita kertoja, vaikka ITU-T määrittelee CS-1:n vain viitekehykseksi älyverkon toteutukselle.

ITU-T:n CS-1

Määriteltäessä älyverkon arkkitehtuurikäsitettä on ITU-T:n CS-1 –suositus ensimmäinen vaihe kohti älyverkkojen standardisoimista. Tämä suositus esittelee CS-1:n antamalla yleisnäkymän älyverkkoihin, sekä kuvaamalla niiden tärkeimmät ominaisuudet ja yleiset toiminnot. [Li93]

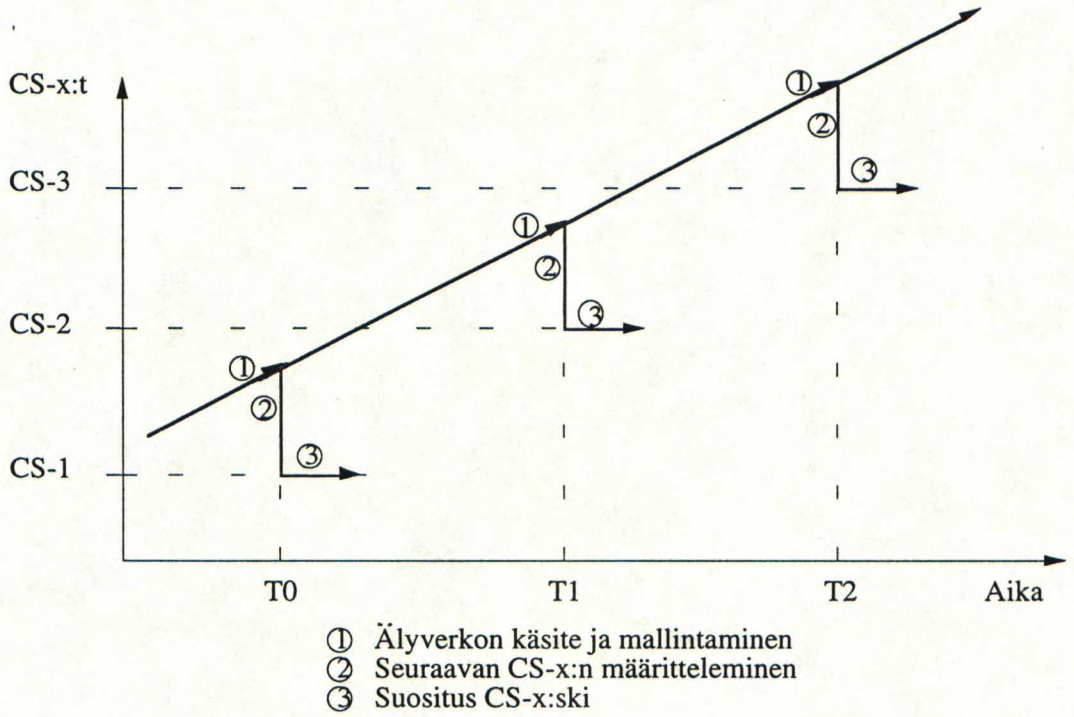
Älyverkkojen standardisoiminen etenee ITU-T:ssä vaiheittain. Älyverkkoarkkitehtuurin käsite määrittelee, että verkon arkkitehtuuriin perustuvia palveluita pystytään määrittelemään ja tuottamaan nopealla aikataululla. Älyverkkojen palvelut käyttävät huomattavan paljon tiedonprosessointitekniikkaa sekä vaativat tehokkuutta verkon resursseilta. Palveluiden toiminnot ovat modulaarisia ja uudelleenkäytettäviä. Niiden luominen ja toteutus ovat integroituja toimintoja. Älyverkkojen toiminnot voidaan joustavasti jakaa fyysisille laitteille.

Nämä toiminnot sisältävät televerkon toiminnoille standardisoidut viestintäliitynnät. Palvelun tilaaja ohjaa joitakin tilatun palvelun ominaisuuksia. Älyverkkoarkkitehtuuria voidaan hyödyntää useassa erityyppisessä verkossa. Tällaisia verkkoja ovat PSTN-, PSPDN-, NMT- ja ISDN-verkot. Saman arkkitehtuurin käyttäminen erilaisissa verkoissa mahdollistaa palvelulogiikan siirtämisen verkosta toiseen. Kuten kuvasta 4 voidaan nähdä jokaisessa eri verkossa palveluiden käyttäminen ja toiminta on yhtenäistä. Palvelut eivät ole riippuvaisia alla olevan verkon laitteista. Verkoissa voidaan tarvittaessa käyttää samoja laitteita, jolloin niiden valmistuskustannukset pienenevät ja verkkojen välinen liityntä yksinkertaistuu. Liitynnän yksinkertaistuminen saa aikaan sen, että monimutkaisten palveluiden tuottaminen on helpompaa. [Col92.]



Kuva 4: Palveluiden tarjoamisen toteutustekniikka [Col92]

Lopullinen älyverkkoarkkitehtuuri on jatkuvasti muuttuva käsite. Jotta älyverkkoarkkitehtuuri hyödyntäisi teknologian kehitystä jollakin tietyllä hetkellä, täytyy sille määritellä tietyt vaiheet, jotka toteutetaan määrityksissä. Tätä kehitystä voidaan esittää seuraavalla kuvalla. [Itu93e]



Kuva 5: CS-x:ien vaiheittaisuus [Itu93e]

Ajatus palvelusta [Itu93e]

Vaikka älyverkkoarkkitehtuuri määritellään siten, että se on tuetuista palveluista riippumaton, on tärkeää, että sen yleiset ominaisuudet kuvataan tarkasti. Ne palvelut ja palvelutoiminteet, joita älyverkkoarkkitehtuurin täytyy tukea, ovat pohjana palveluista riippumattomille lohkoille (Service Independent building Block, SIB), puhelun tilamallille ja palveluiden ohjauksen periaatteille.

Tyyppiä A ja B olevat palvelut [Itu93e]

A-tyypin palvelut, jotka CS-1:ssä kuvataan, voidaan luokitella "yhden pään", "yhden ohjauspisteen" palveluiksi. Muut palvelut ovat B-tyypin palveluita. Tyyppiä A olevien palveluiden ominaisuuksia CS-1:ssä ovat:

1) **Yhden pään palvelutoiminto** käsittää puhelusta yhden ja vain yhden osapuolen. Se on itsenäinen muihin puheluun liittyvien osapuolien näkökulmasta katsoen sekä palvelu- että topologisella tasolla. Itsenäisyys mahdollistaa sen, että toinen yksilö samasta tai toisesta A-tyypin palvelusta voi olla jonkin toisen puheluun liittyvän osapuolen käytössä kunhan palvelutoiminteilla ei ole toistensa kanssa yhteensopivuusongelmia.

2) **Yhden ohjauspisteen toiminto** kuvaa ohjaussuhteen, jossa yksi ja vain yksi palvelun ohjauspiste vaikuttaa puheluun kaikissa sen vaiheissa.

Tyyppiä B olevia palveluita ei CS-1:ssä käsitellä seuraavista syistä:

1) **Toiminnallinen monimutkaisuus** B-tyypin palveluissa käsittää usean älyverkon käyttäjän samanaikaisen liitynnän käytössä olevaan kytkentään. Puhelun aikana käyttäjiä voidaan lisätä tai poistaa vapaasti. Nämä toiminnot tapahtuvat palvelun ohjausfunktion (Service Control Function, SCF) valvomissa, fyysisesti puheluun liittyvissä keskuksissa. Palvelun ohjausfunktio tarvitsee sääntöjä, jotka määrittelevät puheluun osallistuvien tilaajien tunnistamisen. Tämä voi tarkoittaa reaaliaikaista konsultointia niiden eri toimintojen kesken, jotka "edustavat" puhelun eri osapuolia.

2) **Toteutuksen monimutkaisuus** B-tyypin palveluissa voi aiheuttaa lisävaatimuksia palvelun ohjaustoiminnossa olevalle palvelulogiikan ohjelmalle. Ulkopuolisella palvelulogiikalla täytyy olla keskuksen kytkentäresursseista "abstrahoitu" näkymä. Kyseisistä abstrakteista näkymistä on jo olemassa malleja, mutta tällä hetkellä ne ovat vielä teoreettisia ehdotuksia. Keskuksen kytkentäohjelmistoihin tarvitaan huomattavia muutoksia, jotta tällaiset mallit voitaisiin toteuttaa. Verrattaessa A-tyypin palveluita B-tyypin palveluihin voidaan todeta, että A-tyypin palveluiden tarvitsemat muutokset ovat huomattavasti helpompia toteuttaa.

3) **Ohjauksen monimutkaisuus**: verrattuna A-tyypin palveluihin, jossa palvelut kuvataan suhteellisen yksinkertaisella palvelun kytkentätoiminnon ja ohjaustoiminnon välisellä liitynnällä, B-tyypin palvelut ovat toiminnaltaan huomattavasti vaikeampia. Palvelun kytkentätoiminto on "asiakas" ohjaustoiminnon antamalle palvelutietoudelle. Tyyppiä B olevissa palveluissa sen sijaan saatetaan tarvita ohjauksen hajauttamista sekä keskuksen että ulkopuolisen palvelulogiikan välille. Siirtyvässä tietovirrassa täytyy olla useita parametrejä, jotka ohjaavat periaatteessa kahden pisteen välissä olevaa, hajautettua prosessointijärjestelmää.

Nämä monimutkaisuudet toiminnallisuudessa, toteutuksessa ja ohjauksessa A- ja B-tyyppien välillä ovat huomattavia. Tämän vuoksi CS-1 keskittyy tukemaan vain A-tyypin palveluita. On kuitenkin olemassa tilanteita, joissa on mahdollista käyttää A-tyypin älyverkkoteknologiaa joissakin B-tyypin palveluissa.

Palvelu voidaan käsittää joukoksi toiminteita, joita jokin kohde tarjoaa. Näitä toiminteita voivat käyttää myös muut kohteet. Palvelu voi olla joko joukko toiminteita, joita tarjotaan loppukäyttäjälle, kuten konferenssipuhelu tai joukko toimintoja, joita käytetään tuettaessa loppukäyttäjän haluamia palveluita, kuten numeron muunnospalvelu. Kohde voi tukea useita erilaisia palveluita, joilla jokaisella voi olla erilainen liityntä. [Nat92]

Määritelmä 2: Palvelu

> on yksittäinen, kaupallinen ominaisuus, jonka yksi tai useampi ydinpalvelutoiminto määrittelee. Toisia palveluita voidaan käyttää jonkin yksittäisen, erillisen palvelun kanssa, jolloin ne avustavat kyseessä olevaa palvelua.

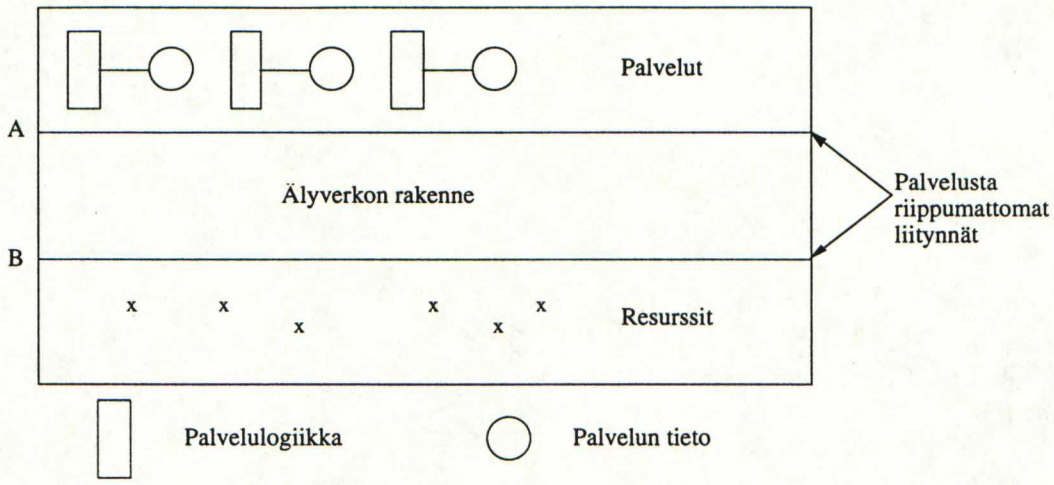
Määritelmä 3: Palvelutoiminne

> on yksittäinen palvelun ominaisuus, jota voidaan käyttää useiden palveluiden ja palvelutoiminteiden yhteydessä. Se on joko palvelun ydin tai valinnaisesti osa sitä.

Tyyppiä A ja B olevat liitynnät [Kun92]

Älyverkot koostuvat kahdenlaisista liitynnöistä, joita ovat tyyppiä A ja B. Tyyppiä A oleva liityntä on lähinnä tarkoitettu ohjelmoijille. Tämä liityntä takaa riippumattomuuden ohjelmien alapuolella olevasta verkosta. Uusia palveluita voidaan näin tarjota nopeasti ilman, että verkon laitteistoa täytyy muuttaa.

Tyyppiä B olevaa liityntää käytetään verkon resurssien ohjaukseen. Tämän liitynnän avulla voidaan yhtenäisellä tavalla ohjata eri valmistajien laitteita. Resurssien ohjaukseen käytetyn liitynnän ollessa standardisoitu pienenevät laitteiden valmistuskustannukset.



Kuva 6: Älyverkkojen looginen järjestys [Kun92]

Toteutus [Li93]

Vuonna 1992 Yhdysvalloissa paikallinen teleoperaattori Bell Atlantic käynnisti AIN 0.1:een perustuvat kaupalliset älyverkkopalvelut. Myös muut Yhdysvaltain teleoperaattorit tarjoavat älyverkkopalveluita ennen vuosikymmenen puoltaväliä.

Euroopassa Deutsche Bundespost Telekom ja PTT Telecom Netherlands ovat kehittämässä yksityiskohtaisia määrityksiä CS-1:n toteuttamiseksi. Näiden määritysten pitäisi olla valmiina ennen vuotta 1995. Myös muut eurooppalaiset teleoperaattorit ovat luomassa älyverkkototeutuksiaan, joihin he siirtävät markkinoilla jo olevat palvelut sekä uudet suunnitteilla olevat palvelumuodot.

Vaikka teleoperaattorit ovat tosissaan sitoutuneet toteuttamaan älyverkkopalveluita, on niiden tarjoamien sovellutusten lukumäärä kehittynyt odotettua hitaammin. Teleoperaattorit ovat olleet varovaisia ja siirtyvät asteittain älyverkkojen toteutukseen.

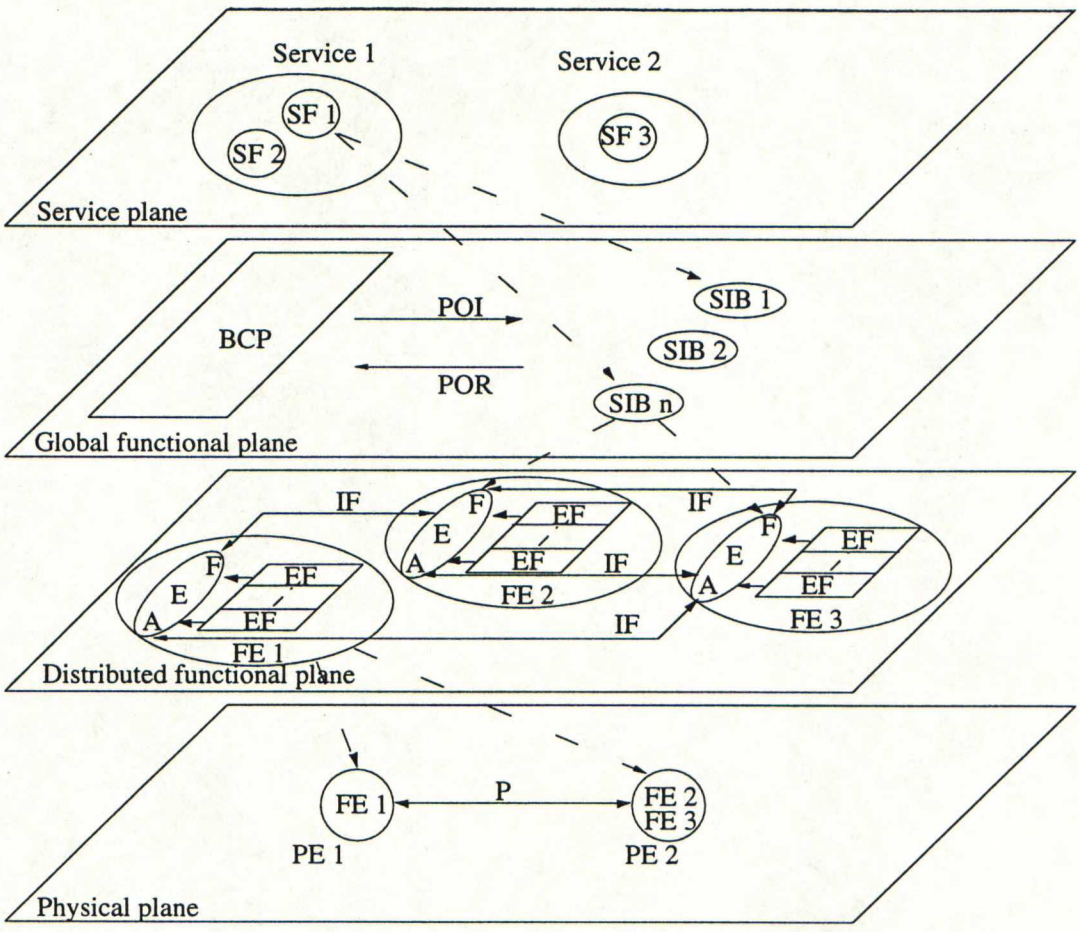
3.1.1 Älyverkkoarkkitehtuurin sisältö [Itu93a]

Älyverkkojen ensimmäinen perusperiaate on tarjota palvelusta riippumattomia toiminteita, "rakennuspalikoita", joita käytetään kehitettäessä uusia palveluita. Tämä rakennuspalikkamaisuus mahdollistaa helpon tavan määritellä ja toteuttaa uusia palveluita. Toinen periaate on riippumattomuus verkon sisäisestä rakenteesta. Sen tarkoituksena on erottaa palvelut tavasta, jolla palveluriippumattomat toiminteet on toteutettu erilaisissa verkoissa.

Toteutustavan riippumattomuus on perustana useille ominaisuuksille. Palvelut voivat käyttää hajautetun verkon eri toimintoja useilla tavoilla. Ne voivat ulottua usean erilaisen verkon alueelle, ollen kuitenkin samalla riippumattomia käyttämiensä verkkojen toteutustavoista. Verkkojen toteutuksessa käytetty tekniikka voi muuttua ilman, että palvelun toiminnalle aiheutuu häiriöitä. Tämä mahdollistaa myös sen, että verkossa käytettävät laitteet voidaan hankkia useilta laitevalmistajilta.

3.1.2 Älyverkkojen käsitteellinen malli, IN Conceptual Model [Itu93a]

Älyverkkojen käsitteellistä mallia ei tulisi käsittää itse arkkitehtuuriksi. Se on vain viitekehys älyverkkojen suunnittelulle ja kuvaukselle. Älyverkkojen standardisoimisessa tullaan käyttämään useita erilaisia malleja ja käsitteitä. Käsitteellisen mallin tarkoituksena on esittää integroitua, muodollista viitekehystä, josta voidaan tunnistaa erilaiset käsitteet ja jossa niitä voidaan kuvata. Olemassaolevia käsitteitä pitää muuttaa siten, että ne sopivat tähän viitekehykseen. Tämän saavuttamiseksi käsitteellinen malli rakentuu neljästä tasosta, jotka kukin esittävät erilaista abstraktia näkymää niihin mahdollisuuksiin, joita älyverkkoon perustuvat verkot tarjoavat. Nämä näkymät kuvaavat älyverkkojen palvelullista näkökulmaa, yleistä toiminnallisuutta, hajautettua toiminnallisuutta sekä palveluiden fyysistä näkökulmaa.



SIB	-	Service Independent building Block	FEA	-	Functional Entity Action
FE	-	Functional Entity	PE	-	Physical Entity
SF	-	Service Feature	EF	-	Elementary Function
IF	-	Information Flow	P	-	Protocol
POI	-	Point of Initiation	POR	-	Point of Return
——►		Pointer	BCP	-	Basic Call Process

Kuva 7: CS:n käsitteellinen malli

3.2 Palvelutaso, "Service Plane" [Itu93b]

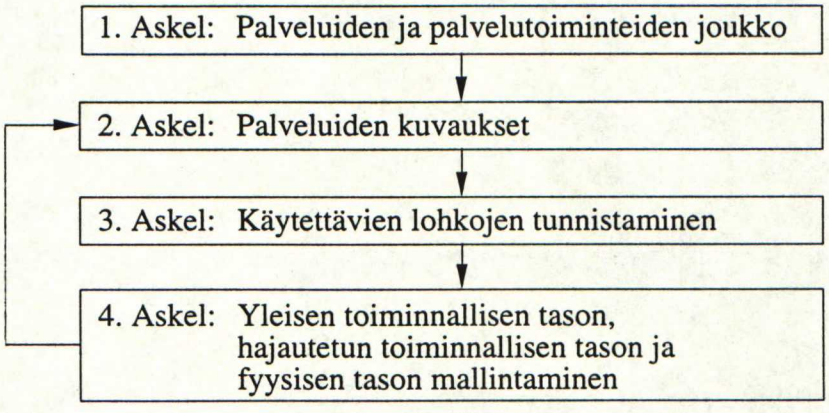
Älyverkon tukemat palvelut voidaan palvelutasolla kuvata loppukäyttäjälle tai tilaajalle yleisinä lohkoina, joita kutsutaan palvelutoiminteiksi (Service Feature, SF). Itse älyverkon palvelulla käsitetään yksittäistä, kaupallisesti saatavilla olevaa toimintoa, jota voidaan kuvata yhdellä tai useammalla toiminteella. Niitä voidaan käyttää myös muiden kaupallisten palveluiden ja palvelutoiminteiden yhteydessä.

Palvelutoiminne on joko itse palvelun ydin tai vaihtoehtoisesti osa palvelua, joka on liitetty jonkin muun palvelun yhteyteen. Palvelutason arkkitehtuuri edustaa puhtaasti palvelusuuntautunutta näkymää. Kaikki, mikä verkon palvelusta on havaittavissa liittyy sen käyttäytymiseen, sellaisena kuin palvelun käyttäjä sen kokee.

3.2.1 Palveluiden kuvaaminen ja palvelun ominaisuuksien vaatimukset

Palveluiden ja palvelutoimintojen kuvaamisella pyritään tunnistamaan palveluista riippumattomat ominaisuudet, joita palveluiden tarjoajat tai verkon operaattorit käyttävät luodessaan ja muuttaessaan palveluita. Esimerkiksi käyttäjän kannalta tärkeitä palveluiden ominaisuuksia ovat puhelun jonottaminen ja asiakaskohtaiset tiedotukset. [Sch92]

On olemassa tarve luokitella strukturoidusti ja tunnistaa erilaiset palvelustariippumattomat ominaisuudet. Kuvassa 8 esitetään korkean tason menetelmä palveluiden analysoimiseksi sekä niiden jakamiseksi palvelustariippumattomiksi lohkoiksi. Nämä uudelleenkäytettävät lohkot muodostavat pohjan yleisen tason sekä hajautetun toiminnallisen tason syötteille. [Itu93b]



Kuva 8: Lähestymistapa palvelustariippumattomien lohkojen analysoimiseksi [Itu93b]

Onkin suositeltavaa, että sellaiset toiminnot, jotka sisältävät toiminnallista mallintamista, käyttäisivät hyväkseen palveluista tehtyä analyysiä, joka perustuu kuvattujen palveluiden vertaamiseen mallinsa kanssa. Tämä vertailu varmistaa sen, että palveluiden tuottamisessa käytetty malli on yhtenäinen. [Itu93b]

3.2.2 Palvelutason mallintaminen [Itu93b]

Palvelut voivat koostua yhdestä tai useammasta palvelutoiminteesta, joka on pienin osa palvelua, minkä käyttäjä voi huomata. Näitä palvelutoiminteita voidaan käyttää myös määriteltäessä ja suunniteltaessa uusia, monimutkaisempia palveluita. Palvelutoiminne koostuu yhdestä tai useammasta palvelustariippumattomasta lohkosta. Kaikki yksittäiset televiestinnän palvelut, joita palvelutasossa tarkastellaan, tulisi kuvata käyttäjän näkökulmasta katsoen siten, että itse palvelun sisäiseen toteutukseen ei oteta kantaa. On tärkeää, että palvelua käytettäessä havaitut ominaisuudet edustavat televiestinnän palveluita.

3.3 Yleinen toiminnallinen taso, "Global Functional Plane" [Itu93c, Itu93g]

Yleinen toiminnallinen taso (Global Functional Plane, GFP) mallintaa verkon toiminnallisuutta yleiseltä tai verkonlaajuiselta näkökannalta. Sellaisenaan älyverkkoarkkitehtuuria tarkastellaan yhtenä oliona yleisessä toiminnallisessa tasossa. Palvelut ja palvelutoiminteet määritellään tässä tasossa niiden verkon toiminteiden avulla, jotka vaaditaan niiden toteuttamiseen. Nämä toiminteet eivät ole palveluista tai palvelutoiminteista riippuvia.

Yleinen toiminnallinen taso sijaitsee palvelutason ja hajautetun toiminnallisen tason välissä. Palvelutasolla määritelty palvelu jaetaan palvelutoiminteisiin, jotka sen jälkeen kuvataan yhdelle tai useammalle yleisessä toiminnallisessa tasossa olevalle palvelusta riippumattomalle lohkolle. Jokainen palvelusta riippumaton lohko kuvataan samalla tavalla yhdelle tai useammalle hajautetussa toiminnallisessa tasossa olevalle toiminnalliselle oliolle (Functional Entity, FE).

Yleiseen toiminnalliseen tasoon sisältyy puhelun lohko (Basic Call Process, BCP), jonka tehtävänä puhelun kytkemisessä on tunnistaa, minkä älyverkon palvelun soittaja haluaa käynnistää. Se sisältää aloituspisteen (Point of Initiation, POI) ja paluupisteen (Point of Return, POR), jotka mahdollistavat yhteyden muodostamisen yleiseen palvelulogiikkaan. Itse palvelustariippumattomat lohkot, jotka ovat normaaleita uudelleen käytettäviä koko verkon kattavia ominaisuuksia, muodostavat palvelut ja palvelutoiminteet.

Yleisen palvelulogiikan (Global Service Logic, GSL) tehtävänä on määritellä, miten palvelusta riippumattomia lohkoja voidaan ketjuttaa siten, että ne kuvaavat palvelutoiminteita. Se kuvaa myös normaalin puhelun lohkon ja palvelusta riippumattomien lohkojen ketjujen välisen yhteyden.

3.3.1 Yleisen toiminnallisen tason mallintaminen

Määritelmän mukaan palvelusta riippumattomat lohkot eivät sisällä tietoutta itse palveluiden toteutuksesta eivätkä täten voi sisältää tietoa muista lohkoista. Siksi yleinen palvelulogiikka on ainoa rakenneosa yleisessä toiminnallisessa tasossa, joka on erityisesti palveluista riippuvainen. Kun älyverkon tukema palvelu aktivoidaan aloituspisteestä laukaisee normaalin puhelun lohko kyseiseen palveluun liittyvän yleisen palvelulogiikan.

Haluttaessa liittää useita lohkoja ketjuksi täytyy niistä tietää käytettävissä olevat liitännät, suorituksen vaihtoehdot sekä niiden sisältämä tieto. Tämän vuoksi yleiseen toiminnalliseen tasoon pitää sisällyttää tieto siitä, miten palvelustariippumattomat lohkot on ketjutettu toisiinsa. Tämä tieto on kuvattu yleisessä palvelulogiikassa. Ketjutettujen lohkojen suorittamisen lopussa yleinen palvelulogiikka kertoo myös, mihin kohtaan normaalia puhelua palvelulogiikan suoritus loppuu määrittelemällä paluupisteen.

Annetulla palvelulla ja palvelutoiminteella pitää olla ainakin yksi aloituspiste. Vaaditusta logiikasta riippuen niillä voi olla useita paluupisteitä.

3.3.2 Palvelustariippumattomat lohkot

Palvelustariippumattomien lohkojen määritelmä

Palvelustariippumaton lohko on yleisellä toiminnallisella tasolla sijaitseva standardi, uudelleenkäytettävä, koko verkon laajuinen ominaisuus, jota käytetään luotaessa palvelutoimintoja. Lohkot ovat luonteeltaan yleisiä, niiden yksityiskohtainen kuvaus sijaitsee hajautetussa toiminnallisessa tasossa sekä fyysisessä tasossa. Ne ovat uudelleenkäytettäviä ja niitä voidaan ketjuttaa toisiinsa erilaisilla yhdistelmillä.

Lohkojen kuvaus

Palvelustariippumattomat lohkot määritellään riippumattomiksi kaikista toiminnallisista ja fyysisistä arkkitehtuureista. Jokaisella lohkolla on yksilöity ja kiinteä liityntä muihin palvelustariippumattomiin lohkoihin. Hajautetun toiminnallisen tason vuorovaikutus toiminnallisten olioiden kanssa on piilotettu yleisellä toiminnallisella tasolla. Yksittäiset lohkot määritellään standardisoidulla tavalla sen vuoksi, että eri laitetoimittajien älyverkkotuotteet pystyisivät tukemaan niitä ja että palveluiden suunnittelijoilla olisi yhtenäinen kuva käytettävissä olevista lohkoista.

Palvelustariippumaton lohko on monoliittinen. Sen yksityiskohtainen toiminnallinen kuvaus on piilotettu sen toteutukseen. Toteutettavat palvelutoiminteet käyttävät yhtä tai useampaa lohkoa. Kuvattavat palvelutoiminteet ovat toteutettavissa rajallisella määrällä lohkoja. Hajautetulla toiminnallisella tasolla palvelusta riippumattomat lohkot kuvataan toiminnallisten olioiden lohkoilla (Functional Entity Action, FEA), jotka voivat sijaita yhdessä tai useassa toiminnallisessa oliossa.

Lohkolla on yksi looginen aloituspiste ja yksi tai useampi looginen paluupiste. Lohkon tarvitsema tieto on määritelty sen omissa parametreissä ja yksittäisen puhelun parametreissä. Lohkot ovat luonteeltaan yleisiä eikä niiden fyysisellä sijainnilla ole merkitystä, sillä koko verkko käsitetään yleisellä toiminnallisella tasolla yhdeksi yhtenäiseksi olioksi.

Lohkojen parametrit

Määritelmän mukaan palvelustariippumattomat lohkot ovat riippumattomia palvelusta ja palvelutoiminteista. Lohkoilla ei ole tietoa edellä olleista tai seuraavista lohkoista. Jotta palvelu voitaisiin määrittää käyttäen palvelustariippumattomia lohkoja, tarvitaan lisäksi jotakin tietoa itse palvelusta. Palvelustariippuva osuus voidaan kuvata käyttäen parametrejä, jotka mahdollistavat lohkojen muokkaamisen sellaisiksi, että niitä voidaan käyttää halutussa palvelussa. Parametrit määritellään jokaiselle lohkolle erikseen ja ne ovat käytettävissä yleisen palvelulogiikan kautta. Palvelusta riippumattomalla lohkolla niitä on kahdenlaisia: yksittäisen puhelun parametrit (Call Instance Data, CID), jotka ovat muuttuvia ja palvelun parametrit (Service Support Data, SSD), jotka ovat pysyviä.

Yksittäisen puhelun parametrit

Yksittäisen puhelun parametrien arvot muuttuvat jokaisessa puhelussa. Niitä käytetään määriteltäessä erityisiä yksityiskohtia, kuten soittajan tai soitetun liittymän puhelinnumeroa. Tämä tieto on saatavilla puhelun tilamallista (esim. soittajan numero), palvelusta riippuvasta lohkosta (esim. muunnettu numero) tai se voi olla käyttäjän antama (esim. soitettu numero tai henkilön tunniste).

Jokaisella yksittäisen puhelun parametrin arvolla on looginen nimi, johon viitataan puhelun parametrin kenttäosoittimella (Call Instance Data Field Pointer, CIDFP). Jos palvelustariippumaton lohko tarvitsee puhelun parametrejä suorittaessaan toimintaansa se saa palvelun parametreistä tiedon puhelun parametrien kenttäsoittimista.

Palvelun parametrit

Palvelun parametrien arvot on yksilöity palvelutoiminteiden kuvauksessa. Kun palvelustariippumaton lohko sisältyy palvelun kuvauksessa olevaan yleiseen palvelulogiikkaan, määritellään logiikassa lohkolle eri palvelun parametrit. Palvelun parametrit koostuvat kahdesta kohdasta.

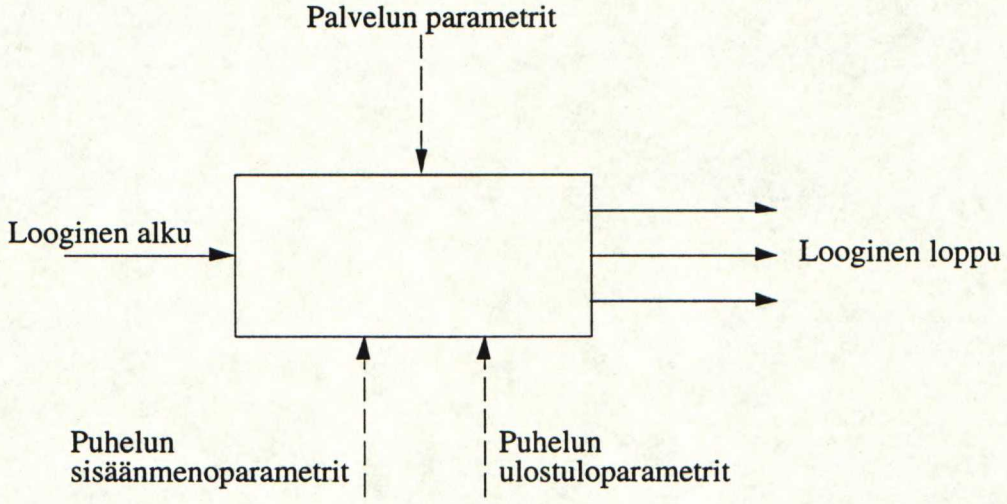
1) **Kiinteiden parametrien** arvot pysyvät muuttumattomina jokaisessa puhelussa. Jos palvelu tai palvelutoiminne on kuvattu käyttäen samaa palvelustariippumatonta lohkoa, määritellään palvelun kiinteät parametrit yksilöllisesti jokaiselle lohkolle

2) **Kenttäosoittimet** yksilöivät, mitä puhelun parametrejä palvelustariippumaton lohko tarvitsee. Samalla määritetään kyseisen tiedon paikallinen sijainti. Määrityksistä käytetään muotoa "CIDFP-xxxx", missä "xxxx" nimeää halutun tiedon. Jos palvelustariippumaton lohko tarvitsee useamman kuin yhden kenttäosoittimen sisältyy palvelun parametreihin useita kenttäosoittimia.

Lohkojen graafinen esitysmuoto

Jokaisella palvelustariippumattomalla lohkolla on yksi looginen sisääntulo ja yksi tai useampia loogisia ulostuloja. Nämä portit on merkitty kuvassa 9 yhtenäisellä nuolella, jotka on sijoitettu elementin vasemmalle ja oikealle sivulle. Jokaisen portin kohdalla on selostus sen toiminnasta. Palvelun parametrit kuvataan elementin

yläpuolella olevalla katkonaisella viivalla piirretyllä nuolella. Samalla tavalla on esitetty puhelun parametrit elementin alapuolella. Puhelun parametrien sisäänmeno- ja ulostuloportit on erotettu toisistaan.



Kuva 9: Palvelustariippumattomien lohkojen graafinen esitysmuoto

Puhelun lohko

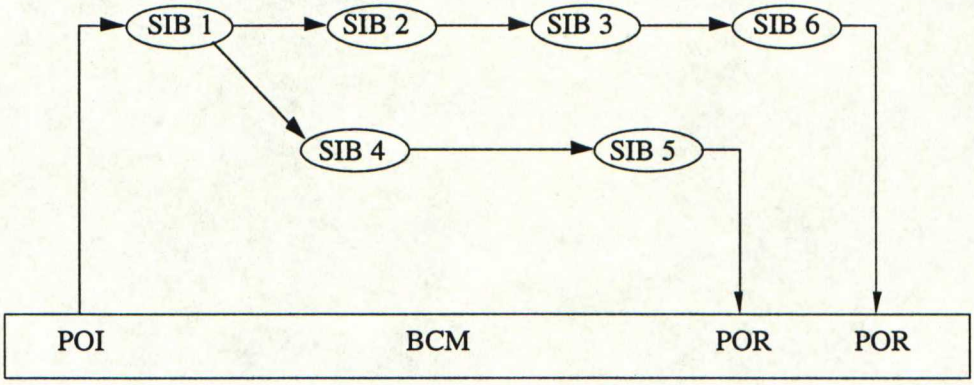
Puhelun lohko vastaa normaalista yhteyden muodostamisesta kahden liittymän välille. Sen voidaan ajatella olevan erikoistunut palvelustariippumaton lohko, jonka perusominaisuutena on yhteyden luominen, sen purkaminen ja puhelun parametrien tallentaminen muuta käyttöä varten.

Älyverkkojen tukemat palvelut ja palvelutoiminteet kuvataan käyttämällä puhelun lohkoon liitettyjä, ketjutettuja, palvelustariippumattomia lohkoja. Liitynnät puhelun lohkon ja palvelustariippumattomien lohkojen välillä muodostetaan aloitus- ja paluupisteissä.

Nämä pisteet määritellään seuraavasti:

1) **Aloituspiste** (POI) on se kohta puhelun lohkossa, josta palvelustariippumattomien lohkojen toiminta käynnistetään

2) **Paluupisteiden** (POR) lukumäärä määritellään analysoimalla palveluilta vaadittavat ominaisuudet.



Kuva 10: Esimerkki yleisen toiminnallisen tason mallintamisesta

Erillisen aloitus- ja paluupiste -toiminteen esittely on tarpeellista, sillä eri kohdista puhelun lohkoa käynnistetyt samanlaiset palvelustariippumattomat lohkot voivat edustaa erilaisia palveluita. Myös samat palvelustariippumattomat lohkot, jotka käynnistetään samasta kohtaa puhelun lohkoa, mutta jotka lopettavat toimintansa eri kohtiin puhelun mallia, voivat edustaa erilaista palvelua.

3.3.3 Yleinen palvelulogiikka, "Global Service Logic"

Yleinen palvelulogiikka voidaan määritellä siten, että se on se "liima", joka määrittelee palvelussaolevien ketjutettujen lohkojen suoritusjärjestyksen. Jokainen yleinen palvelulogiikka on (mahdollisesti) yksilöllinen jokaiselle yksittäiselle puhelulle. Tosin joitakin yhteisiä kohtia on havaittavissa. Tälläisia yhteyksiä on puhelun lohkon aloitus- ja paluupisteissä, käytetyissä palvelustariippumattomissa lohkoissa, loogisissa yhteyksissä lohkojen välillä, joissa yhteyksiä voi olla sekä lohkojen että puhelun lohkon vuorovaikutuspisteiden välillä, sekä lohkoille määritellyissä sisäänmeno- ja ulostuloparametreissä.

3.4 Hajautettu toiminnallinen taso, "Distributed Functional Plane" [Itu93d, Itu93h]

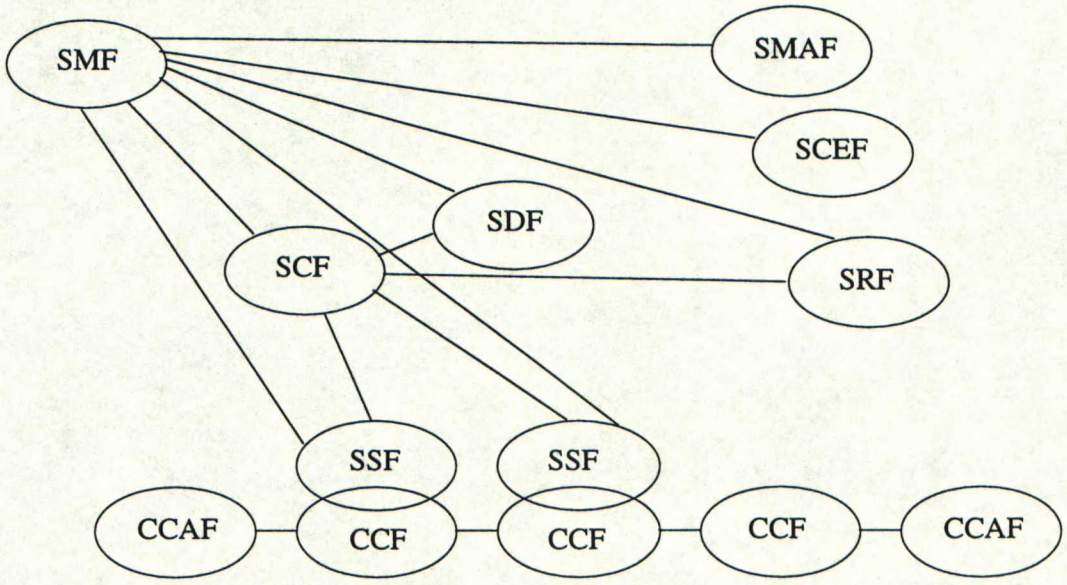
Älyverkkojen hajautetun toiminnallisen tason (Distributed Functional Plane, DFP) vaatimuksissa ja oletuksissa sen arkkitehtuurin tulee olla yhtenäinen älyverkon toiminnallisessa mallissa määritellyn viitekehyksen kanssa. Sen tulee määritellä ne elementit ja riippuvuussuhteet, jotka ovat tarpeellisia tuettaessa älyverkkojen tarkoituksia. Hajautetun toiminnallisen tason arkkitehtuuri mallintaa niitä toimintoja, joita suoritetaan älyverkoissa. Sen tulee mahdollistaa joustava tuki laajalle joukolle

palveluita myös tulevaisuudessa, kun älyverkkojen ratkaisut muuttuvat. Hajautetun toiminnallisen tason arkkitehtuurin tulee olla riippumaton laitetoimittajista ja toteutustavoista, jolloin se mahdollistaa joustavuuden useassa erilaisessa verkkorakenteessa eikä se aseta mitään vaatimuksia kansallisten verkkojen rakenteille. Aluksi hajautetun toiminnallisen tason arkkitehtuuri määrittelee palvelun suorituksen ominaisuudet. Kun palveluiden luominen ja palveluiden ja verkon hallinnan ominaisuudet on saatavilla lisätään ne hajautetun toiminnallisen tason arkkitehtuurin määrittelyihin.

3.4.1 Toiminnalliset oliot

Toiminnallinen olio on yhdessä paikassa oleva yksittäinen joukko toiminteita. Yksi tai useampi toiminnallinen olio voi sijaita samassa fyysisessä laitteessa. Kaksi erillistä toiminnallista oliota sisältävät toisistaan erillisiä toiminteita. Niissä voi olla myös yksi tai useampi toiminne, joka on samanlainen molemmille oliolle. Yhtä toiminnallista oliota ei voida jakaa kahden fyysisen laitteen kesken, vaan se pitää kuvata kokonaisuudessaan yhdelle fyysiselle laitteelle. Jokaiseen toiminnalliseen olioon liitetään yksilöllinen tunniste, eikä niiden kuvaukset sisällä sellaista huolto- ja ylläpitotoiminteita, jotka eivät suoranaisesti liity palveluiden tarjoamiseen. Fyysinen sijainti ei ole ainoa peruste, jonka mukaan toiminteita yhdistetään toiminnallisiksi olioiksi. Ryhmittelemisessä pitää myös ottaa huomioon kaikki teknologian ja kustannustekijöiden asettamat ehdot.

Kuvassa 11 olevat toiminnalliset oliot muodostavat kommunikoivia pareja, joita yhdistävät informaatiovuot. Toiminnallisten olioiden riippuvuudet toisistaan on merkitty kaaviossa olioiden välisillä viivoilla. Riippuvuuksilla on määrittelyt, jotka yksilöivät mallissa olevat informaatiovuot. Samanlainen riippuvuuden määrittely voi esiintyä toiminnallisessa mallissa useammin kuin kerran. Jos joidenkin toiminnallisten olioiden välillä ei esiinny yhteyttä mallissa, ei niiden välistä riippuvuussuhdetta ole määritelty. Jos kommunikoiva pari on sijoitettu erilleen toisistaan, erillisiin fyysisiin laitteisiin, määrittelee niiden välinen riippuvuussuhde tiedonsiirrossa vaadittavan protokollan.



Kuva 11: Ehdotus CS:n toiminnalliseksi arkkitehtuuriksi

3.4.2 Älyverkon palveluihin liittyvät toiminnalliset oliot

Puhelun ohjaustoiminnefunktion (Call Control Agent Function, CCAF) tehtävänä on antaa käyttäjälle liityntä puhelinverkkoon käyttäjän antamien ohjeiden mukaisesti. Vaadittaessa se luo, muokkaa ja purkaa puhelun tai palveluliittymän. Sen tehtävänä on antaa liittymä CCF:ään käyttäen palvelupyyntöjä luodakseen, muokatakseen ja purkaakseen puhelun tai palveluliittymän. Funktion toimintoja on lisäksi CCF:n lähettämien viestin vastaanottaminen ja niiden välittäminen käyttäjälle. Lisäksi sen tehtävänä on ylläpitää tietoa puhelun tai palvelun tilasta niin kuin se sitä itse havannoi.

Puhelun ohjausfunktio (Call Control Function, CCF) suorittaa verkossa tapahtuvaa yhteyden prosessointia ja ohjausta. Se muodostaa, muokkaa ja purkaa palveluliittymiä CCAF:n pyyntöjen mukaisesti. Sen toimintoihin kuuluu myös liittää ja yhdistellä CCAF:ään niitä toiminnallisia olioita, jotka osallistuvat tiettyyn palveluliittymään. Sen hallinnassa on palveluliittymässä käytettyjen CCAF:ien toiminnallisten olioiden riippuvuudet. Se mahdollistaa liipaisumekanismin käytön, jolla voidaan hyödyntää älyverkon toimintoja. Sitä voidaan hallita, päivittää ja muuten ohjata älyverkkoon liittyvissä toiminnoissa SCEF:stä käsin.

Palvelun kytkentäfunktio (Service Switching Function, SSF) yhdessä SCF:n kanssa antaa käytettäväksi joukon toimintoja, joita tarvitaan niiden välisessä liitynnässä. Se laajentaa SCF:n logiikkaa siten, että logiikka huomaa SCF:n liipaisun ja että CCF:lle muodostuu yhteys SCF:n kanssa. Funktio valvoo CCF:n ja SCF:n välistä merkinantoa. Sen tehtävänä on muokata SCF:n alaisuudessa olevien älyverkkopalveluiden pyyntöihin liittyviä palveluliittymän prosessointitoimintoja. Sitä voidaan hallita, päivittää ja muuten ohjata älyverkkoon liittyvissä toiminnoissa SCEF:stä käsin.

Palvelun ohjausfunktio (Service Control Function, SCF) suorittaa älyverkkopalveluiden prosessoinnissa palvelun ohjausta. Se voi olla vuorovaikutuksessa muiden toiminnallisten olioiden kanssa saadakseen lisää tietoa ja saattaakseen loppuun palvelulogiikan suorituksen. Se liittyy ja on yhteydessä CCF:n, SSF:n, SRF:n ja SDF:n kanssa. Funktio sisältää älyverkkopalvelun tarvitseman logiikan ja prosessointikapasiteetin. Se pystyy tarvittaessa ottamaan yhteyttä ja viestimään muiden televerkossa olevien SCF:n kanssa. Sitä voidaan hallita, päivittää tai muuten ohjata älyverkkoon liittyvissä toiminnoissa SCEF:stä käsin.

Palvelun tietofunktio (Service Data Function, SDF) sisältää tietoa asiakkaasta sekä hänen käyttämästään verkosta. Suorittaessaan palvelulogiikan ohjelmaa SCF:llä on reaaliaikainen yhteys SDF:ään. SDF on tarvittaessa yhteydessä ja vuorovaikutuksessa SCF:n kanssa ja se voi olla yhteydessä muiden SDF:ien kanssa. Sitä voidaan hallita, päivittää ja muuten ohjata älyverkkoon liittyvissä toiminnoissa SCEF:stä käsin.

Palvelun resurssifunktio (Service Resource Function, SRF) sisältää erikoistuneita resursseja, joita älyverkko tarvitsee palvelua suoritettaessa. Se on yhteydessä ja vuorovaikutuksessa SCF:n ja SSF:n kanssa. SRF voi sisältää käyttäjältä saadun tiedon vastaanottamiseen ja lähettämiseen liittyvää logiikkaa ja prosessointia. Se voi sisältää samankaltaista toiminnallisuutta, jota on SCF:ssä silloin kun puhelu yhdistetään resursseihin. Sitä voidaan hallita, päivittää ja muuten ohjata älyverkkoon liittyvissä toiminnoissa SCEF:stä käsin.

Palvelun luonti- ja testausfunktio (Service Creation Environment Function, SCEF) sisältää määrittelyjä, joita hyväksikäyttämällä älyverkossa oleva palvelun tuottaja voi rakentaa haluamansa palvelun. Funktio on toteutettu siten, että se näyttää palvelun tuottajalle yksinkertaistetun mallin palvelun suorituksesta, jolloin palvelulogiikan luominen helpottuu. SCEF voi tarvittaessa hallita kaikkia älyverkon toiminnallisia olioita.

Palvelun hallintafunktio (Service Management Function, SMF) mahdollistaa älyverkon palveluiden hallinnan. Se voi myös osallistua palvelun käytön ohjaukseen. SMF:n tärkeimpiä tehtäviä on ohjata SCF:n ja SDF:n välistä yhteyttä. Hallinnan kohteena on SCF:n lähettämien laskutus- ja laskutustietojen ohjaaminen oikeisiin osoitteisiin ja palveluun liittyvän tiedon ohjaaminen SDF:lle.

Palvelun hallinnan ohjausfunktio (Service Management Access Function, SMAF) antaa palvelun tuottajalle liittymän SMF:ään. Se mahdollistaa palveluiden ohjauksen SMF:n kautta.

3.5 Yleisen toiminnallisen tason kuvaaminen hajautetulle toiminnalliselle tasolle [Itu93d]

Yleisen toiminnallisen tason kuvauksen hajautetulle toiminnalliselle tasolle pitää olla yhtenäinen älyverkon käsitteellisen mallin kanssa. Kuvauksessa jokaiseen palvelusta riippumattomaan lohkoon täytyy liittyä ainakin yksi toiminnallinen olio.

Puhelun tilamalli (Basic Call State Model, BCSM) on korkean tason äärellinen automaatti, jossa kuvataan niitä CCF:n toimintoja, joita käytetään muodostettaessa sekä ylläpidettäessä käyttäjän haluamia yhteyksiä. Puhelun tilamalli yksilöi CCF:ssä olevat puheluun liittyvät perustoiminnot ja kuvaa, miten nämä toiminnot puhelun prosessoinnissa liitetään toisiinsa.

Puhelun tilamalli yksilöi ne puhelun kytkennässä olevat kohdat, joissa älyverkkopalvelun logiikka voi liittyä siihen. Erityisesti se luo viitekehyksen, jossa voidaan kuvata ne puhelun mallin ja kytkennän toiminnot, joista voidaan aktivoida älypalvelu. Tämä mallinnetaan kuvaamalla pisteet, jossa nämä tapahtumat ovat havaittavissa. Puhelun tilamallin osia ovat puhelupisteet (Points in Call, PIC), havaintopisteet (Detection Points, DP), siirtymät ja tapahtumat. Puhelupisteet määrittelevät CCF:n tapahtumat, jotka vaaditaan, kun halutaan suorittaa yksi tai useampi älyverkon palvelulogiikan ohjelma. Havaintopisteet osoittavat niitä kohtia

puhelussa ja kytkennässä, joissa ohjauksen siirto on mahdollista. Siirtymät osoittavat normaalia puhelun ja kytkeytymisen siirtymistä puhelupisteistä toiseen. Tapahtumat saavat aikaan siirtymiä puhelupisteiden välillä.

3.6 Fyysinen taso, "Physical Plane" [Itu93e, Itu93i]

Älyverkon käsitteellisen mallin fyysinen taso määrittelee verkossa olevat erilaiset fyysiset oliot, sekä yhteydet näiden olioiden välillä. Fyysisen tason arkkitehtuurin tulee olla yhtenäinen käsitteellisen mallin kanssa. Älyverkon käsitteellinen malli on työkalu, jota voidaan käyttää suunniteltaessa älyverkkoarkkitehtuuria.

3.6.1 Vaatimukset

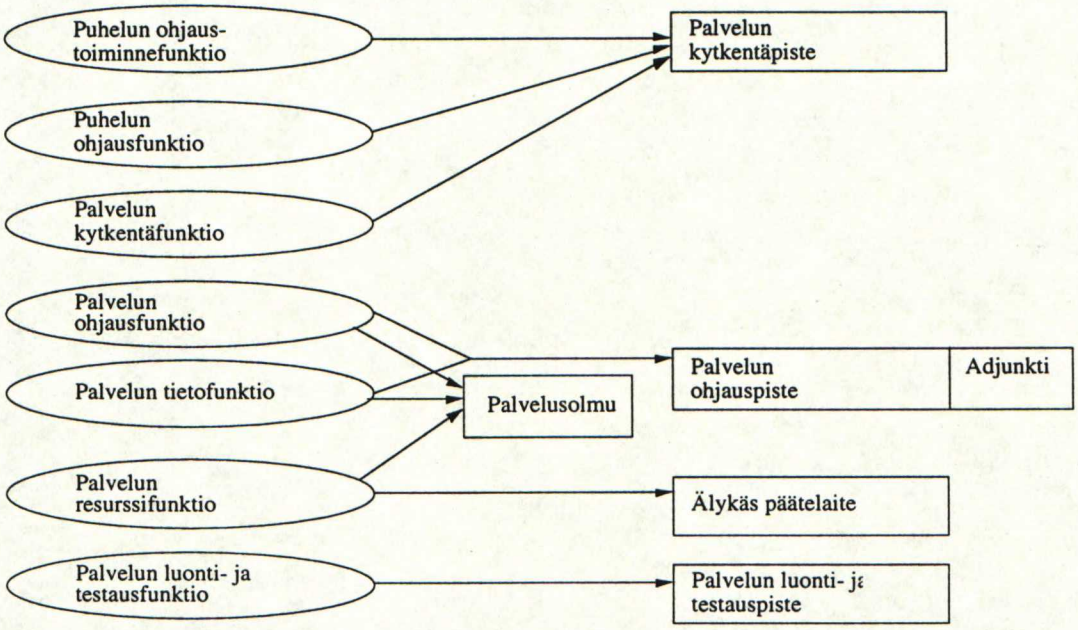
Hajautetun toiminnallisen tason oliot kuvataan fyysisiksi laitteiksi, jolloin yksi tai useampi toiminnallinen olio voidaan kuvata samalle laitteelle. Yksittäistä toiminnallista oliota ei voida jakaa kahden laitteen kesken, vaan se täytyy kuvata vain yhdelle laitteelle. Sama toiminnallinen olio voidaan kuvata kuitenkin usealle laitteelle, ei kuitenkaan kahta samanlaista oliota samaan laitteeseen. Fyysiset laitteet voidaan koota yhdeksi laitekokonaisuudeksi, jolloin ne voivat muodostaa yksittäisen, yhtenäisen liitynnän palvelun tarjoajille. Laitevalmistajien tulee kehittää laitteita toiminnallisten olioiden kuvauksista sekä standardisoiduista liittymistä,. Heidän tulee lisäksi tukea sekä yleisesti käytettyä tekniikkaa että uutta tekniikkaa sitä mukaa kun sitä kehitetään.

3.6.2 Oletukset

Älyverkon käsitteellistä mallia käytetään kehitettäessä älyverkon fyysistä arkkitehtuuria. Olemassa olevaa ja uutta teknologiaa voidaan käyttää fyysisten laitteiden kehitystyössä. Hajautetun toiminnallisen tason määrittelemät toiminnalliset oliot ja fyysisen tason standardisoitu liittymä tekee verkosta laitevalmistajista riippumattoman sekä palveluistariippumattoman. Riittävä määrä liityntöjä täytyy määritellä palvelujen tukemiselle ja niiden luomiselle.

3.6.3 Fyysiset laitteet

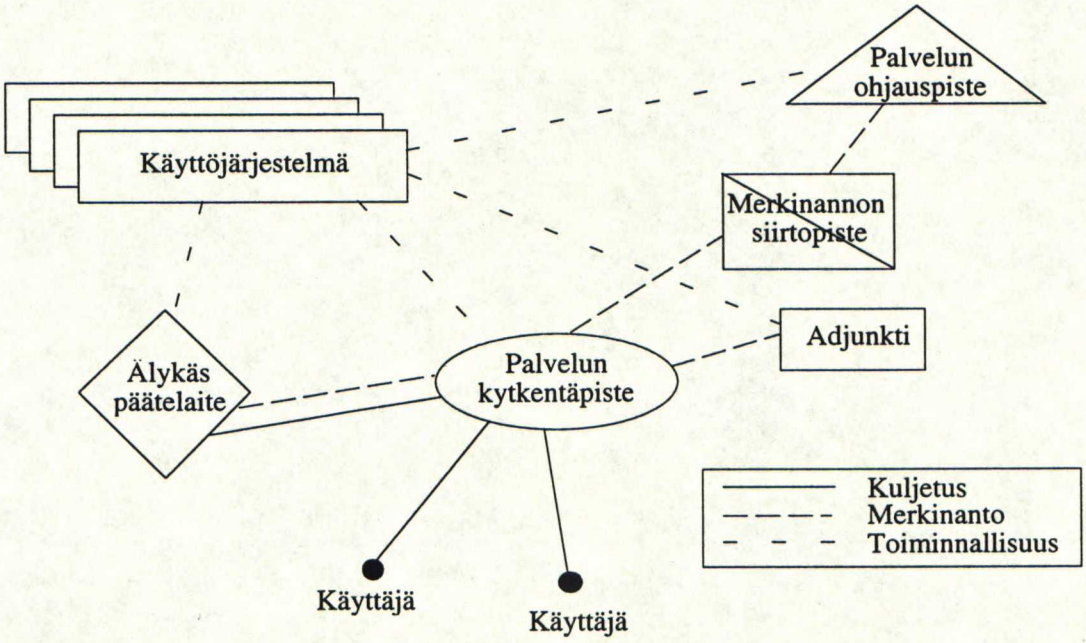
Älyverkkojen fyysinen arkkitehtuuri käsittää solmuja ja liityntöjä. Arkkitehtuurissa olevat toiminnot täytyy sisällyttää fyysisiin tuotteisiin, joita teleoperaattorit käyttävät luodessaan ja ylläpitäessään palveluita. Fyysinen laite voi sisältää yhden tai useamman älyverkon toiminteen. Se voi olla sulautettu ohjelma tai täydellinen laite- tai ohjelmistojärjestelmä. [Li93]



Kuva 12: Eräs vaihtoehto kuvata älyverkon toiminnot fyysisille laitteille [Li93]

Kuten kuvasta 12 nähdään eri fyysiset laitteet voivat sisältää saman toiminnallisen olion. Tämä on mahdollista siksi, että vaikka kaikki älyverkon palvelut toteutetaan kytkentäjärjestelmän ja palvelulogiikan välillä on niiden siirtämiseksi laitetasolla olemassa useita erilaisia ratkaisuja. [Li93]

Fyysisen tason laitteet



Kuva 13: Esimerkki älyverkon laitteiden sijoittumisesta [Joh92]

Palvelun kytkentäpiste (Service Switching Point, SSP) on julkisen puhelinverkon keskus, joka antaa käyttäjälle liittymän verkkoon ja suorittaa palvelun tarvitsemat kytkentätoimenpiteet. Samalla se mahdollistaa myös liitynnän älyverkkoon. Kytkentäpistettä on muutettu siten, että se pystyy tunnistamaan älyverkon palvelupyynnöt. Se pystyy viestimään muiden laitteiden kanssa ja vastaamaan niiden tekemiin kyselyihin. Toiminnallisesti SSP sisältää puhelun ohjausfunktion, palvelun kytkentäfunktion ja jos se on paikallinen keskus, myös puhelun ohjaustoiminnefunktion. Se voi myös valinnaisesti sisältää palvelun ohjausfunktion, palvelun resurssifunktion tai palvelun tietofunktion. [Li93]

Palvelun ohjauspiste (Service Control Point, SCP) vastaa palvelulogiikasta ja puhelun ohjauksesta. Se sisältää palvelulogiikan ohjelman (Service Logic Program, SLP), jota käytetään älyverkkojen palveluissa. Siinä voi myös valinnaisesti olla asiakkaan tuottamaa tietoa. Useat palvelun ohjauspisteet voivat sisältää saman logiikkaohjelman sekä tiedot, joilla voidaan taata palveluiden häiriötön toimiminen sekä kuormituksen tasainen jakautuminen. Toiminnallisesti palvelun ohjauspisteessä on palvelun ohjausfunktio. Se voi valinnaisesti sisältää myös palvelun tietofunktion. Palvelun ohjauspiste voi hakea tietoa palvelun tietopisteeltä joko suoraan tai merkinantoverkon kautta. Tietopiste voi olla samassa verkossa ohjauspisteen kanssa tai se voi sijaita toisessa verkossa. Palvelun ohjauspiste voi

olla yhden tai useamman merkinannon siirtopisteen (Signalling Transfer Point, STP) kautta yhteydessä muiden palvelun ohjauspisteiden, palvelun kytkentäpisteiden tai älykkäiden päätelaitteiden kanssa. Yhteydenpito tapahtuu merkinantoverkon avulla. [Li93]

Palvelun tietopiste (Service Data Point, SDP) sisältää logiikkaohjelman käyttämän tiedon. Tätä tietoa hyväksikäyttämällä luodaan yksittäiset palvelut. Palvelun tietopiste sisältää palvelun tietofunktion. Palvelun ohjauspiste ja palvelun hallintapiste voivat olla yhteydessä siihen suoraan tai merkinantoverkon kautta. Palvelun tietopiste voi olla yhteydessä myös muissa verkoissa olevien tietopisteiden kanssa. [Li93]

Älykäs päätelaite (Intelligent Peripheral, IP) on laitteisto, joka ohjaa ja hallitsee resursseja. Se mahdollistaa joustavan liittymän käyttäjän ja verkon välille. Vaihtoehtoisesti resursseihin liityttäessä käytettävä kytkentämatriisi voi olla ulkopuolisen logiikkaohjelman käytettävissä. Esimerkkejä erilaisista resursseita ovat ilmoitusten asiakaskohtaistaminen ja yhdistäminen, puheen syntetisointi ja tunnistaminen, äänitaajuusmerkkien kerääminen, konferenssisillat, äänigeneraattorit sekä protokollamuuntimet. [Li93]

Älykäs päätelaite sisältää palvelun resurssifunktion ja valinnaisesti palvelun kytkentäfunktion ja puhelun ohjausfunktion. Näitä käytetään kun haluttaan antaa ulkopuolinen liityntä päätelaitteessa oleviin resursseihin. Päätelaite voi olla yhteydessä yhden tai useamman palvelunkytkentäpisteen kanssa. [Li93]

Adjunkti (Adjunct, AD) on toiminnaltaan palvelun ohjauspisteen kaltainen (se sisältää samat toiminnalliset oliot), mutta fyysisesti se on kytketty suoraan palvelun kytkentäpisteeseen. Adjunktin ja palvelun kytkentäpisteen välinen viestintä tapahtuu nopean väylän kautta. Tämä järjestelmä mahdollistaa sen, että adjunktin vasteaika on huomattavasti tavallista palvelun ohjauspistettä nopeampi. Adjunkti voi olla kytkettynä useampaan kuin yhteen palvelun kytkentäpisteeseen ja palvelun kytkentäpiste voi olla kytkettynä useampaan kuin yhteen adjunktiin. [Li93]

Palvelusolmu (Service Node, SN) voi ohjata älyverkon palveluita ja se voi olla joustavasti yhteydessä käyttäjän kanssa. Se viestii suoraan yhden tai useamman palvelun kytkentäpisteen kanssa. Sillä on suora merkinanto- ja tiedonsiirtoyhteys jokaisen palvelun kytkentäpisteen kanssa. Toiminnallisesti palvelusolmu sisältää palvelun ohjausfunktion, palvelun tietofunktion, palvelun resurssifunktion ja palvelun kytkentäfunktion tai puhelun ohjausfunktion. Palvelusolmussa oleva

palvelun kytkentäfunktio tai puhelun ohjausfunktio on erittäin läheisesti kytketty laitteessa olevaan palvelun ohjausfunktioon eivätkä ulkopuoliset ohjausfunktiot voi käyttää sitä hyväkseen. [Li93]

Samoin kuin adjunkti, palvelusolmussa oleva palvelun ohjausfunktio vastaanottaa viestejä palvelun kytkentäpisteeltä. Se suorittaa palvelun logiikkaohjelmia ja lähettää viestejä palvelun kytkentäpisteelle. Siinä olevaa logiikkaohjelmaa voidaan kehittää samoilla palvelun kehitysmenetelmillä, joilla niitä kehitetään palvelun ohjauspisteille ja adjunkteille. Palvelusolmussa oleva palvelun resurssifunktio mahdollistaa samankaltaisen toiminnan, joka on älykkäällä päätelaitteella. [Li93]

Palvelun kytkentä- ja hallintapiste (Service Switching and Control Point, SSCP) on yhdessä paikassa oleva yhdistetty palvelun ohjaus- ja kytkentäpiste. Toiminnallisesti se sisältää palvelun ohjausfunktion, tietofunktion, puhelun ohjausfunktion ja palvelun kytkentäfunktion. Se saattaa sisältää myös palvelun resurssifunktion toimintoja. [Li93]

Palvelun luonti- ja testauspiste (Service Creation Environment Point, SCEP) on palvelun määrittelyyn, kehittämiseen ja testaamiseen käytetty järjestelmä. Sen avulla palvelun tarjoaja, tulevaisuudessa myös palvelun tilaaja ja käyttäjä, voi luoda logiikkaohjelmia ilman, että hänen tarvitsee tietää palvelun toteutuksen yksityiskohtia. Liityntä puhelun prosessointiin tapahtuu kytkentämallin toiminnallisten olioiden ja liipaisupisteiden kautta. Toiminnallisesti se sisältää palvelun luonti- ja testausfunktion. Sillä on suora yhteys palvelun hallintapisteeseen. [Li93]

Palvelun hallintapiste (Service Management Point, SMP) suorittaa palvelun hallinnollisia toimintoja, palvelun tarjonnan toimintoja sekä palvelun sijoittumisen ohjausta. Esimerkkejä sen toiminnoista ovat tietokantojen hallinta, verkon tarkkaileminen ja testaaminen, verkon liikenteen seuraaminen sekä tiedon kerääminen. Toiminnallisesti se sisältää palvelun hallintafunktion ja valinnaisesti myös palvelun hallintatoiminnefunktion tai palvelun luonti- ja testausfunktion. Palvelun hallintapiste voi olla yhteydessä kaikkien verkossa olevien laitteiden kanssa. [Li93]

Palvelun hallinnan ohjauspiste (Service Management access Point, SMAP) antaa valituille käyttäjille, kuten palvelun tarjoajille ja joillekkin tilaajille, liittymän SMP:lle. Se on hyvin läheisessä yhteydessä SMP:n kanssa. SMAP:ia käyttämällä voidaankin hallita useita SMP:tä. Toiminnallisesti se sisältää SMAF:n. [Itu93e]

4 KÄYTETTY TYÖKALU [Mal93]

4.1 Yleistä

Työssä käytetyn työkalun nimi CVOPS tulee sanoista 'C-based Virtual Operating System'. Työkalu on Valtion Teknillisen Tutkimuskeskuksen teletekniikan laboratoriossa protokollien kehitykseen ja testaukseen kehitetty ohjelmisto. Sen pääkäyttäjiin kuuluvat erilaiset tietoliikenneprotokollia kehittävät ja testaavat teleliikenneyritykset. Ohjelmiston suosioon on vaikuttanut sen sopivuus useisiin eri laiteympäristöihin ja käyttöjärjestelmiin, tehokkuus, ylläpidettävyys sekä saatavilla oleva laaja aineisto erilaisia protokollien toteutuksia.

4.2 Ohjelman käyttämät tiedostot

Ohjelma sisältää laajennetun äärellisten automaattien kuvaukseen (Extended Finite State Automaton, EFSA) käytettävän kielen, jota C-kielen lisäksi käytetään protokollien kuvaamisessa. Tyypillinen protokollan toteutus sisältää tiedostoja, jotka on kirjoitettu äärellisten automaattien kielellä ja tiedostoja, jotka on kirjoitettu C-kielellä. Tiedostot erotetaan toisistaan loppuliitteillä (*.str, *.if, *.in, *.aut, *.c sekä *cod.c). Kaikkia tiedostoja ei välttämättä tarvita kaikissa toteutuksissa, sillä joissakin tapauksissa voidaan kahden tiedoston sisältö yhdistää yhdeksi. Tiedon jakaminen tiettyihin tiedostoihin helpottaa mahdollisesti myöhemmin tapahtuvaa toteutuksen tarkastelua.

*.str

Protokollan eri kerrosten väliset liitynnät kuvataan tiedostossa, jonka loppuliite on "*.str". Tiedostossa luetellaan kaikki protokollassa olevat kerrokset sekä ne liitynnät, joilla yksittäinen kerros voi kommunikoida. CVOPS lataa ohjelmiston käynnistyksen aikana tiedoston muistiin, jonka jälkeen se luo tiedoston sisältämän tiedon perusteella protokollan määrittelemät automaatit.

*.if

Protokollan kahden kerroksen välinen liityntä kuvataan tiedostossa, jonka loppuliite on "*.if". Tässä tiedostossa luetellaan kerrosten väliset primitiivit sekä parametrilohkot, jotka sisältävät primitiivien parametrit. Tätä tiedostoa käytetään kahdella tavalla: tiedosto C-käännetään, koska se on osana C-tiedostoja ja CVOPS käyttää sitä ohjelmiston suorituksen aikana.

*.in

Tämä tiedosto sisältää kerroksen sisäisen rakenteen. Kuten edellä tätä tiedostoa käytetään kahdella tavalla: tiedosto C-käännetään ja CVOPS käyttää niitä ohjelman suorituksen aikana symbolitaulujen alustamiseen ja suorituksen seurantaan. Tiedosto sisältää kerroksen protokollan tietoyksiköiden (Protocol Data Unit, PDU) määrittelyt, automaatin tilojen nimet, sen käyttämät muuttujat ja ajastimet.

*.aut

Protokollan CVOPS-toteutus sisältää sekä äärellisen automaatin kielisiä, että C-kielisiä kuvauksia. Tässä tiedostossa kuvataan protokollan kerrosten automaatit, jotka ovat muodoltaan joukko tila-syöte-toiminta -kolmikoita. Automaatin suorittaessa toimintojaan se voi käyttää edellisissä tiedostoissa määriteltyjä palveluprimitiivejä, protokollan tietoyksiköitä, muuttujia, ajastimia, jne. Tiedosto voi sisältää myös joitakin funktioita palveluprimitiivien parametrien alustamiseen ja käsittelemiseen.

*.c

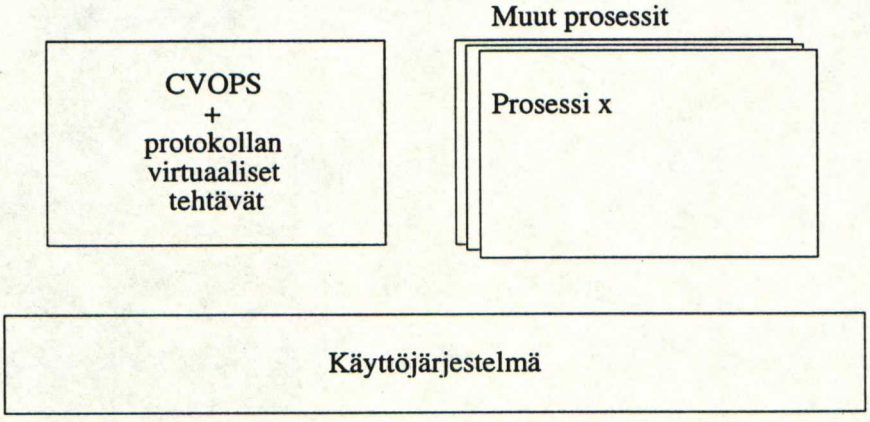
Tiedosto sisältää kerroksen käytössä olevat C-kieliset funktiot, joita äärellisen automaatin lauseet voivat käyttää hyväkseen. Se sisältää myös joitakin funktioita palvelun primitiivien parametrien alustamiseen ja käsittelemiseen.

*cod.c

Sisältää protokollan tietoyksiköiden koodaukseen ja dekoodaukseen käytetyt funktiot.

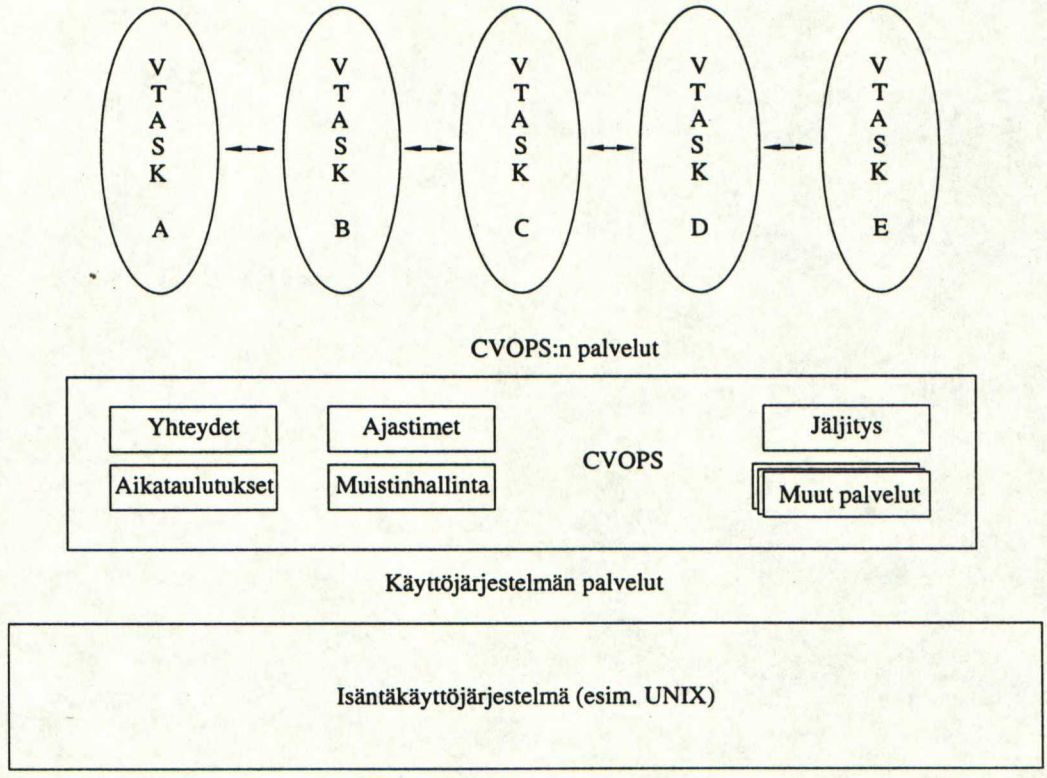
4.3 Virtuaalisten tehtävien rakenne

Protokollatason toteutuksen pienintä rakennetta kutsutaan virtuaaliseksi tehtäväksi (virtual task, VTASK). Virtuaaliset tehtävät kommunikoivat keskenään lähettämällä viestejä käyttäen CVOPS:in tukemaa sanomajärjestelmää.



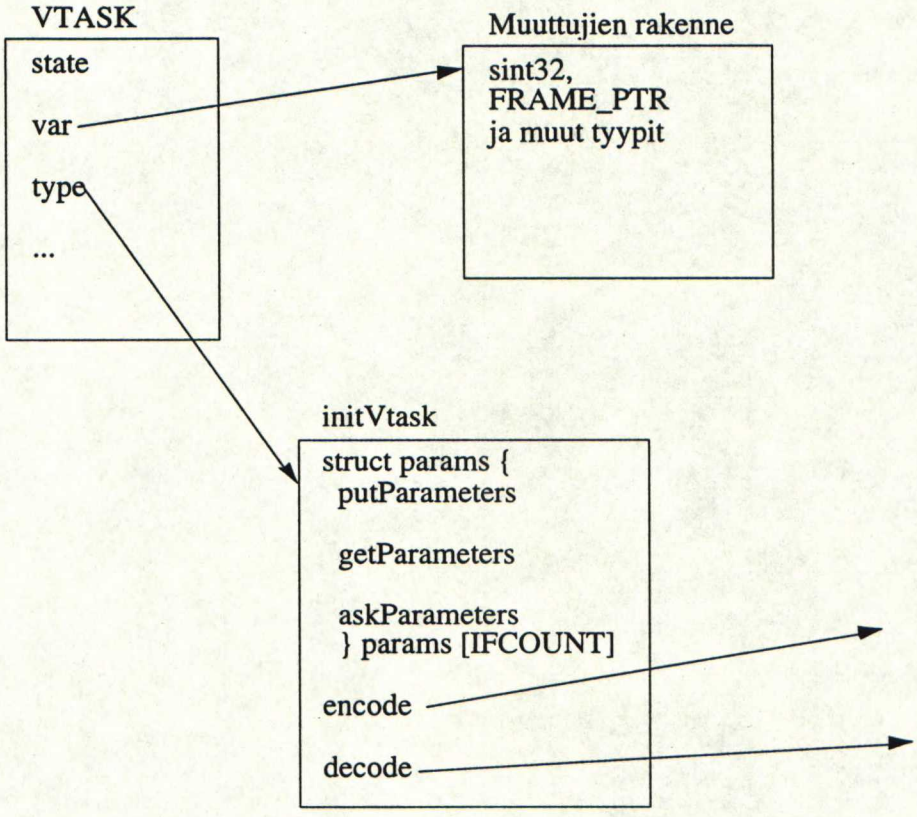
Kuva 14: CVOPS ja virtuaaliset tehtävät moniajoympäristössä

CVOPS on sekä protokollan toteuttamiseen käytetty kehitystyökalu että niiden suoritusympäristö. Viimeisteltyyn toteutukseen sisältyy osa protokollien kehitykseen ja testaamiseen käytettyä järjestelmää. Tavallisesti CVOPS-järjestelmää suoritetaan jonkin isäntäkoneen käyttöjärjestelmän päällä. Tällöin CVOPS erottaa protokollan toteutuksen isäntäkoneen käyttöjärjestelmästä; virtuaaliset tehtävät käyttävät vain CVOPS:n tarjoamia palveluita. Tarvittaessa CVOPS suorittaa kutsuja isäntäkoneen käyttöjärjestelmälle. Protokollan toteutus saadaan tällöin riippumattomaksi laiteympäristöstä, jolloin niiden siirtäminen eri laitteistoille on vaivatonta. Protokollan toteutuksen täytyy esimerkiksi ajastimen käynnistämiseksi käyttää CVOPS:n tukemia palveluita sen sijaan, että se tekisi suoria kutsuja isäntäkoneen käyttöjärjestelmän ajastinpalveluille.



Kuva 15: Virtuaalisen käyttöjärjestelmän käsite

Isäntäkoneen käyttöjärjestelmän suorittaessa muistissa olevia ohjelmia ja toteuttaessa niiden tekemiä palvelupyyntöjä, tekee vastaavasti virtuaalinen käyttöjärjestelmä saman virtuaalisille tehtäville. Kaikki virtuaaliset tehtävät ovat C-kielisiä rakenteita, jotka ovat tyyppiä VTASK. Muuttujat, ajastimet, jne. eivät voi suoraan sisältyä virtuaalisen tehtävän rakenteeseen, koska yksityiskohdat, kuten muuttujien lukumäärä, riippuvat käsiteltävänä olevan protokollan kerroksesta. Tämän vuoksi virtuaalisen tehtävän kentät ovat osoitinkenttiä, jotka viittaavat kerroksessa oleviin rakenteisiin.



Kuva 16: Virtuaalisen tehtävän rakenne

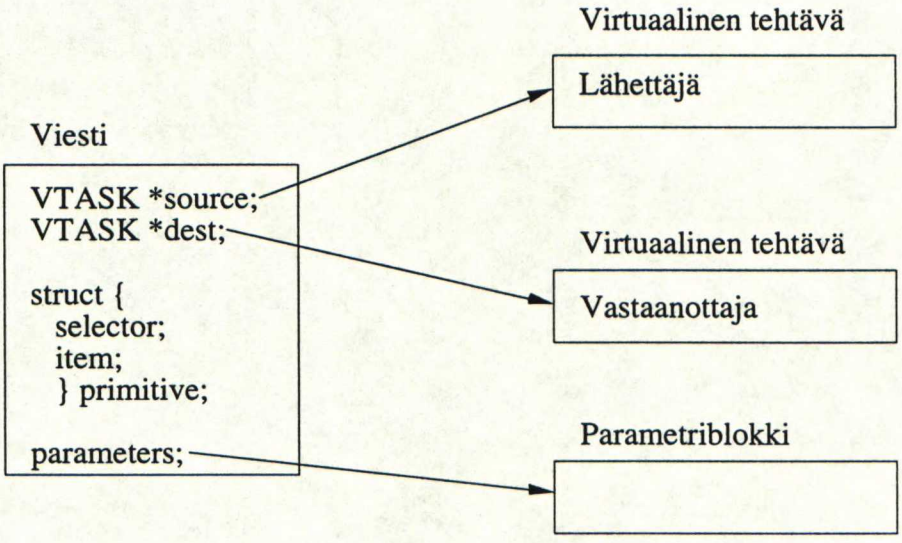
Muuttujien rakenteeseen viitataan **var**-kentän osoittimella. Automaatissa voidaan muuttujiin viitata suoraan niiden nimillä, mutta C-kielisessä osuudessa täytyy käyttää osoittimia, jotka viittaavat haluttuun muuttujaan. Protokolla voi käyttää useita samantyyppisiä virtuaalisia tehtäviä. Nämä viittaavat jaettuun rakenteeseen, joka sisältää tyypille ominaista tietoa, esimerkiksi osoittimen funktioihin, joihin äärellisen automaatin kielestä voidaan viitata. Virtuaalisen tehtävän rakenteessa oleva **type**-kenttä on osoitin tyypin rakenteeseen. Tyypin rakenne voi olla jaettu kaikille samantyyppisille virtuaalisille tehtäville.

4.4 Virtuaalisten tehtävien viestintä ja liitynnät

Virtuaaliset tehtävät kommunikoivat keskenään lähettämällä viestejä CVOPS:in sanomanvälitysjärjestelmän kautta. Kun virtuaalinen tehtävä lähettää palveluprimitiivin tai protokollan tietoyksikön, luo CVOPS automaattisesti viestin rungon. Viestin luomisen jälkeen siinä olevat kentät täytetään ja se laitetaan viestijonoon. Myöhemmin protokollan suorituksessa CVOPS ottaa viestin jonosta ja suorittaa vastaanottavan automaatin ohjelman. Vastaanotettua viestiä käytetään tilasiirtymän syötteenä.

CVOPS:n viestin tärkeimmät osat ovat:

1. dest: osoitin vastaanottajaan
2. source: osoitin lähettäjään
3. primitive: primitiivin (protokollan tietoyksikön) tunnistin, kenttä sisältää liitynnän tunnuksen (valitsijan) ja primitiivin tunnuksen
4. parameters: osoitin parametrilohkoon



Kuva 17: Viestin rakenne

Virtuaalisella tehtävällä voi olla useita liityntöjä. Se voi lähettää ja vastaanottaa viestejä vain liityntöjensä kautta. Protokollan toteuttaja ei voi keksiä omia nimiä liitynnöille. Käytettävät nimet on määritelty seuraavasti: 'UP', 'DOWN', 'PEER', 'MGMT', 'RES1', 'RES2', 'RES3, jne. Nimet 'UP', 'DOWN', 'PEER' ja 'MGMT' on valittu kuvaamaan toteutuksien kerroksellisuutta, RESx-nimiä voidaan käyttää muissa liitynnöissä.

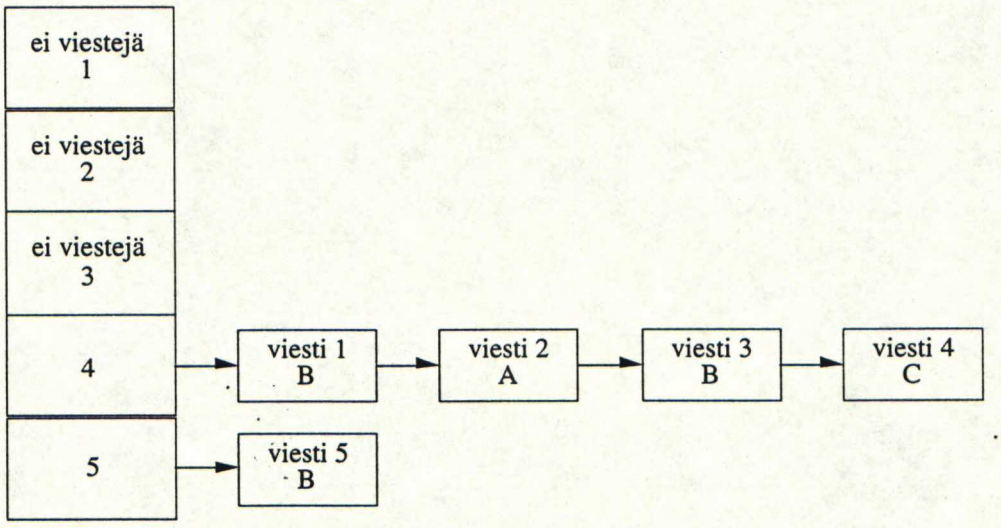
Voidakseen käsitellä viestin CVOPS:n täytyy tietää sen vastaanottaja. Tyypillisesti lähettäjällä on osoitin ylempänä ja alempana oleviin kerroksiin. Protokollan toteuttajan ei tarvitse tietää, mihin nämä osoittimet on tallennettu. Kun virtuaalinen tehtävä lähettää primitiivin, se antaa lähetyksen yhteydessä halutun liitynnän nimen, esimerkiksi UP. Tätä nimeä CVOPS käyttää osoittimena ylempään kerrokseen.

4.5 Järjestelmän käyttäminen testauksessa

Kun viesti on lähetetty CVOPS tallentaa sen viestijonoonsa. Viestijonoja voi olla useita, joilla kaikilla voi olla erilainen kiireellisyysaste. Yhdessä jonossa voi olla useita viestejä. Viestiä käsitellessään CVOPS:n suorittamat perusoperaatiot ovat:

1. Ota viesti korkeimman prioroteetin omaavasta, ei-tyhjästä jonosta.
2. Suorita viestin vastaanottavan virtuaalisen tehtävän ohjelma. Vastaanottavan tehtävän tilasiirtymä voi sisältää uusien viestien lähettämisen, jotka liitetään järjestelmän viestijonoon.
3. Jos jonossa ei ole muita viestejä siirry kohtaan 1.
4. Puskurit ovat tyhjinä. Odota ulkopuolista syötettä (muilta käytössä olevilta ohjelmilta, verkosta tai näppäimistöltä) tai ajastimen laukeamista. Kun ulkopuolelta saadaan syöte tai ajastin laukeaa luo viesti ja pistä se jonoon.
5. Siirry kohtaan 1.

Käytetyin viestien prosessointialgoritmi perustuu virtuaalisten tehtävien jakamiin viestijonoihin. Koska jonot ovat yhteisiä kaikille virtuaalisille tehtäville ja viestit suoritetaan FIFO-järjestyksessä, viestin vastaanottaja ei saa sitä ennen kuin sekä korkeamman prioriteetin viestijonot että aikaisemmin lähetetty, saman prioriteetin omaamat viestit on toimitettu perille. Kuvassa 8 virtuaalinen tehtävä "C" ei saa viestiä numero 4 ennenkuin "B" on saanut viestit 1 ja 3 sekä "A" viestin 2.



Kuva 18: Jaetut viestijonot

Toinen viestien prosessointitapa on erottaa viestin välitys ja suorituksen aikatauluttaminen toisistaan. Tässä menetelmässä jokaisella virtuaalisella tehtävällä on oma viestijononsa. Tämä tapa on usein suositumpi kuin ensimmäisenä kuvattu silloin, kun siirrettävä tieto toteutettavassa protokollassa täytyy jakaa useaan erikseen lähetettävään osaan.

4.5.1 Automaattien ohjaaminen

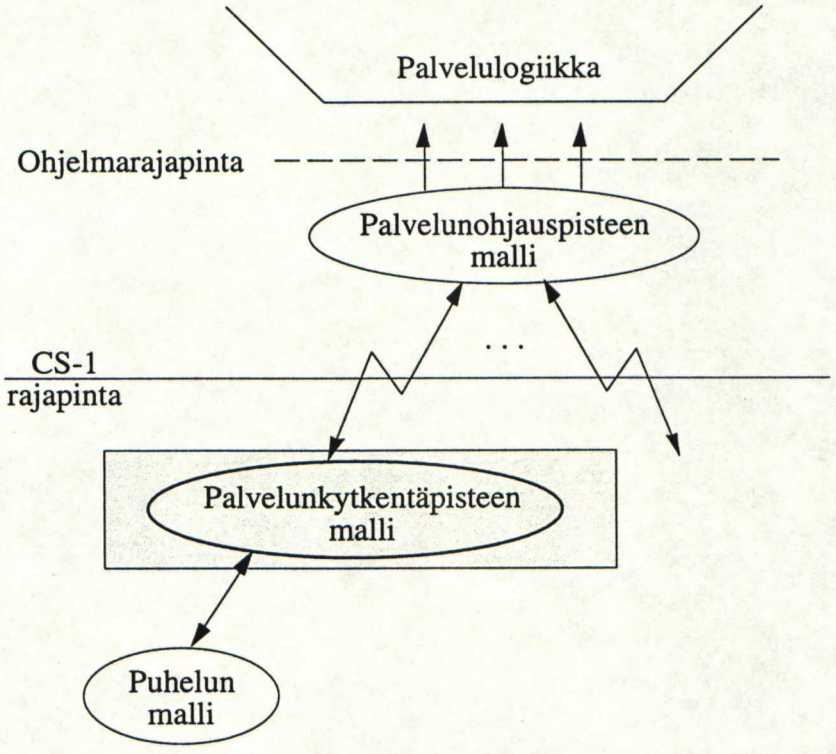
Toteutetussa järjestelmässä olevia automaatteja ohjataan CVOPS:n komentojen avulla. Järjestelmän käyttäjä voi lähettää automaateille viestejä suoraan tai niin halutessaan hän voi muodostaa useista erillisistä viesteistä makroja, jotka automatisoivat järjestelmän käyttöä.

CVOPS tukee myös automaattien tilojen sekä lähetettyjen viestien lähempää tarkastelua. Haluttaessa tutustua tarkemmin automaattien toimintaan CVOPS muodostaa luettelon automaattien sisältämien muuttujien arvoista sekä kaikista niiden lähettämistä ja vastaanottamista viesteistä. Järjestelmän käyttäjä voi itse valita, miten yksityiskohtaisesti järjestelmää seurataan. Jokaiselle automaatille voidaan yksityiskohtaisesti rajata seurattavat ominaisuudet. Seurantaa tulisi käyttää vain silloin, kun halutaan testata toteutettua järjestelmää, sillä sen käyttäminen hidastaa CVOPS:n toimintaa huomattavasti.

5 SIMULAATTORITYÖKALUN TOTEUTUS

5.1 Yleistä

ITU-T:n CS-1 –suositukseen perustuvan simulaattorin toteutuksessa keskitytään palvelun kytkentäpisteen simulointiin (kuvassa 19 tummennettu alue). Palvelunohjauspisteen osuus toteutetaan erikseen.



Kuva 19: Toteutuksen laajuus

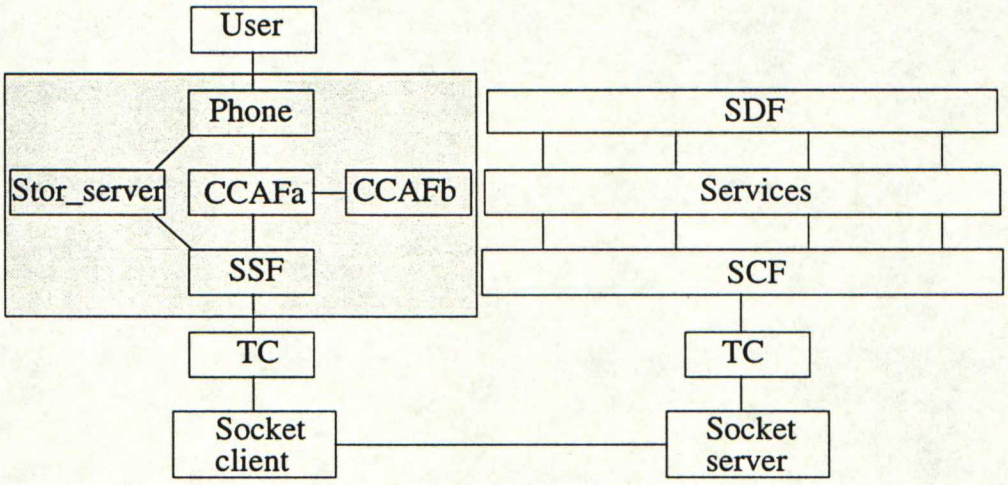
Diplomityön pohjana käytetään Telecom Finland Oy:n Telekehityskeskuksessa toteutettua Nokia Telecommunications Oy:n määrittelemää älyverkon sovellusprotokollaan pohjautuvaa ratkaisua palvelun kytkentäpisteen simulaattorityökalusta. Olemassa oleva järjestelmä toimii mallina CS-1:een perustuvalle simulaattorityökalulle.

Simulaattorityökalun toteutus on syytä jakaa useaan erilliseen kerrokseen, jotka kukin voidaan toteuttaa yksitellen ylhäältä alaspäin edeten. Laajennettaessa simulaattorityökalua ensimmäisenä tehtävänä on muuttaa käytössä olevaa järjestelmää siten, että sen automaattien rakenteet vastaavat toteutettavaa järjestelmää. Ensimmäisessä vaiheessa muunnetaan automaattien toisilleen

lähettämät sanomat CS-1:en kaltaisiksi. Toisessa vaiheessa luodaan puhelun mallit kaikille verkon komponenteille. Kolmannessa vaiheessa muunnetaan automaattien lähettämien viestien välittämät parametrit vastaamaan oikeata puhelutilannetta. Neljännessä vaiheessa koodataan lähetettävät viestit ASN (Abstract Syntax Notation) -koodauksella. [Kar93]

CS-1:ssä määritetyt toiminnalliset oliot voidaan kuvata fyysisiksi olioiksi usealla eri tavalla. Verkko-operaattorit ja laitteiden valmistajat voivat määrittää, miten toiminnalliset oliot parhaiten sijoitetaan laitteisiin. Operaattoreilla ja laitevalmistajilla voi kuitenkin olla erilaisia näkemyksiä vaihtoehtojen paremmuudesta, minkä vuoksi suosituksessa otetaan huomioon myös hajautetut vaihtoehdot.

5.2 Automaatit



Kuva 20: Älyverkkosimulaattorityökalun yleinen arkkitehtuuri

Simulaattorityökalu koostuu useasta erillisestä automaatista, jotka viestivät keskenään käyttäen ITU-T:n CS-1 –protokollaa. Käytetyt automaatit ovat rakenteeltaan lähellä deterministisiä äärellisiä automaatteja [Lew81]. Ne eroavat äärellisistä automaateista siten, että ne voivat syötteen lisäksi myös itse lähettää sanomia. Yksittäisen automaatin suorittamat toiminnot riippuvat vastaanotetusta viestistä sekä automaatin senhetkisestä tilasta. Kun kaikki viestin sisältämät toiminnot on suoritettu voi automaatin tila olla sama, missä se oli vastaanottaessaan viestin tai sen tila voi olla muuttunut. Automaatin uusi tila riippuu edellisessä tilassa vastaanotetusta viestistä. Jokainen automaatti on toteutettu siten, että se pystyy kaikissa tilanteissa käsittelemään sille lähetetyt viestit. Koska kaikkia automaatin viesti-tila –yhdistelmiä ei ole järkevää määritellä, voidaan joitakin viestejä yleistää

siten, että niille luodaan yleisiä sääntöjä, jotka ovat voimassa automaatin jokaisessa tilassa. Kun automaatti vastaanottaa viestin, jolle sillä ei senhetkisessä tilassa ole käsittelysääntöä, käyttää se tällöin viestin tulkitsemiseen yleisempää sääntöä.

Kuvassa 20 oleva varjostettu alue osoittaa ne järjestelmässä olevat automaatit, jotka on saatettu ITU-T:n CS-1 –protokollan mukaisiksi. Lisäksi on ollut tarpeellista muokata palveluiden automaatteja siten, että ne pystyvät käsittelemään CS-1 –protokollan mukaisia viestejä. Muu osa kuvasta on toteutettu aikaisempien projektien yhteydessä.

5.2.1 Puhelimen automaatti

Kuvaus

Puhelimen automaatti on yksinkertainen, kahden tilan automaatti, jolla simuloidaan puhelinlaitteen toimintaa. Se antaa käyttäjälle mahdollisuuden simuloida puhelimen käyttöä älyverkkoarkkitehtuuriin perustuvassa järjestelmässä.

Toiminnot

Puhelimen automaatti muodostaa simuloidun puhelinliitynnän älyverkon kytkentäkeskukseen. Tämän liittymän kautta voidaan simulointiympäristössä käyttää erilaisia älyverkkopuheluita. Kuten oikeassakin puhelimessa on automaatilla kolme päätoimintoa.

1) **Automaatin "luuri" voidaan nostaa ylös**, jolloin käyttäjä ilmaisee automaatille halusta asettaa puhelu johonkin toiseen puhelinverkon liittymään.

2) **Automaatille syötetään numeroita**, silloin kun käyttäjä haluaa ilmaista sen puhelinliittymän numeron, johon puhelu yhdistetään. Toisin kuin oikeassa puhelimessa, ei puhelimen automaatti välitä näppäilystä saamaansa merkkiä välittömästi eteenpäin, vaan se odottaa kunnes käyttäjä on syöttänyt puhelinnumeron kokonaisuudessaan.

3) **Automaatin "luuri" voidaan laskea alas**, kun käyttäjä haluaa lopettaa käynnissä olevan puhelu.

Liitynnät

Puhelimen automaatilla on liitynnät sekä ylös- että alaspäin. Ylöspäin olevalla liitynnällä simuloidaan puhelimen toimintoja käyttäjälle. Käyttäjä voi ohjata puhelimen automaattia silloin, kun CVOPS on komentotilassa. Jos CVOPS on suorittamassa käyttäjän ohjaamia toimintoja tai omia sisäisiä hallintatoimenpiteitään täytyy käyttäjän odottaa kunnes CVOPS on suorittanut tehtävät ja palannut komentotilaan.

Alaspäin olevilla liitynnöillä puhelimen automaatti viestii IP:n ja CCAF:n kanssa. Puhelimen automaatti lähettää puhelun tilamallin alussa keräämänsä numerot CCAF-automaatille jatkokäsittelyä varten. Puhelimen automaatti on yhteydessä IP:n kanssa, kun älyverkon palvelulogiikka haluaa antaa käyttäjälle ohjeita tai kun se haluaa käyttäjän reagoivan antamiinsa ohjeisiin. Tavallisimmin käyttäjältä haluttu vaste on numerotietoa, esimerkiksi salasanan näppäileminen.

5.2.2 Puhelun ohjauspisteen automaatti

Kuvaus

CCAF automaatti on jakaantunut kahteen erilliseen osaan CCAFa:han ja CCAFb:hen. Näistä ensimmäinen simuloi puhelun ohjauspistettä puhelun tilamallin käynnistävässä päässä ja jälkimmäinen sen loppupäässä.

Toiminnot

CCAFa on yksinkertaistettu malli oikeasta CCAF:stä. Automaatin toimintoihin kuuluisivat puhelinliittymän käyttöoikeuden tarkistaminen, tiedon kerääminen puhelinliittymältä, kerätyn tiedon analysoiminen sekä reitityksen valitseminen. Näitä ominaisuuksia ei ole toteutettu simulaattorin tässä vaiheessa. Muutoin ohjauspisteen toimintoja on pyritty mallintamaan totuudenmukaisesti.

Kuten CCAFa on CCAFb:kin yksinkertaistettu malli CCAF:stä. Automaatin toimintoihin kuuluisi tarkistaa, onko kyseiseen liittymään oikeus reitittää tulevia puheluita. Tätä ominaisuutta ei ole toteutettu simulaattorin tässä vaiheessa. Lisäksi automaatilta puuttuu yhteys puhelimen automaatille, mikä mahdollistaa puhelun tilamallin kevyemmän mallintamisen. Muutoin ohjauspisteen toimintoja on pyritty mallintamaan totuudenmukaisesti.

Liitynnät

Alaspäin olevalla liitynnällä CCAFa viestii SSP:n automaatin kanssa. Se välittää puhelimen automaatilta saamansa numerotiedot suoraan SSP:n käsiteltäviksi. SSP:n ohjeiden mukaisesti se ohjaa puhelun kytkemistä. Sivulle olevan liittymän kautta se viestiin CCAFb:n kanssa. Tämän liittymän kautta se saa tiedon soitetun numeron tilasta.

CCAFb:llä on liitynnät sekä ylöspäin, että sivulle CCAFa:han. Ylöspäin olevalla liitynnällä ohjataan suoraan puhelinliittymän toimintoja ilman, että välissä olisi puhelimen automaattia. Sivulla olevalla liittymällä CCAFb viestii CCAFa:n kanssa. Tämän liittymän kautta se saa tiedon puhelinliittymälle tulevista viesteistä.

5.2.3 Palvelun kytkentäpisteen automaatti

Kuvaus

SSP suorittaa palveluiden kytkennän ohjausta. Se voi muodostaa useita samanaikaisia älyverkkopalveluihin kohdistuvia puheluita. Jokainen otettu puhelu luo yhteyden CCAF-automaatin ja SSP:n automaatin välille. Yksittäiseen puheluun voi samanaikaisesti liittyä enemmän kuin yksi palvelu. Kunkin yksittäisen puhelun ohjauksesta huolehtii kuitenkin vain yhden palvelun logiikka.

Toiminnot

SSP päättelee CCAFa:lta saamastaan informaatiosta, tuleeko sen suorittaa kysely SCP:lle. Jos käyttäjän antaman puhelinnumeron neljä ensimmäistä numeroa vastaavat sen tiedossa olevien älyverkon numeroihin lähettää se viestin puhelusta SCP:lle. Muutoin se lähettää CCAFa:lle tiedon, että käyttäjän antama puhelinnumero on tavallinen puhelinnumero. Jos SCP tarvitsee lisätietoja puhelusta pyytää SSP tarvittavan tiedon tyypistä riippuen joko CCAFa:ltä tai IP:ltä.

Liitynnät

Ylöspäin olevalla liitynnällä SSP viestii CCAFa:n automaatin kanssa. Se välittää CCAFa:lta tulevat älyverkkopuhelut alhaalla olevien kerrosten kautta SCP:lle. Alaspäin suuntautuva viestintä koodataan TC-kerroksen ymmärtäviksi sanomiksi, joiden perille saapumisesta huolehtii socket-kerros. Merkkitiedon keräämistä varten SSP:llä on lisäksi yhteys IP:hen, jonka kautta se saa käyttäjältä SCP:n tarvitsemaa tietoa.

5.2.4 Älykkään päätelaitteen automaatti

Kuvaus

IP:n automaattia käytetään, kun halutaan antaa järjestelmän käyttäjälle palveluun liittyviä viestejä ja kun käyttäjältä halutaan saada vastauksia esitettyihin kyselyihin. Tätä automaattia käytetään kahdessa erilaisessa tarkoituksessa.

1) **Käyttäjän ohjaamisessa** erilaisissa valinta- ja virhetilanteissa, jolloin näyttöruudulle kirjoitetaan oikeassa palvelussa kuultavat ääniviestit.

2) **Kysyttäessä käyttäjältä numerotietoja**, jolloin käyttäjän syöte voi olla yksittäinen numero (esimerkiksi erilaisissa valintatilanteissa) tai numerosarja (esimerkiksi henkilökohtainen tunnusluku).

Toiminnot

Saatuaan SSP:ltä viestin suorittaa automaatti viestissä olleet komennot. Komentoina voi olla tiedotuksen kirjoittaminen käyttäjän näyttöpäätteelle tai numeron tai numeroiden kysyminen käyttäjältä.

Liitynnät

Alaspäin olevalta liittymältä IP saa toimintaohjeet, jotka SSP on sille lähettänyt. Saamiensa ohjeiden mukaisesti se tulostaa näyttöpäätteelle SCP:n haluamat tiedotukset. Ylöspäin olevalta liitynnältä IP voi pyytää puhelimen automaattia keräämään käyttäjältä SCP:n tarvitsemaa merkkitietoutta.

6 JOHTOPÄÄTÖKSET

Simulaattorityökalun toteutuksessa on otettu huomioon sille määritellyt vaatimukset. Se simuloi älyverkossa tapahtuvia palvelupyyntöjä simulaattorityökalun käyttäjälle. Työssä käytössä ollut laitteisto on asettanut rajoituksia ratkaisun laajuudelle. Ohjelman vaste käyttäjälle ei pyri olemaan aidon puhelun tuntuinen. Simulaattorilla onkin tässä vaiheessa tarkoituksena vain testata palveluiden suorituksen oikeata kulkua.

6.1 Kehitysnäkymät

Luotu ohjelma ei itseasiassa simuloi oikeata puhelinverkkoa kovinkaan hyvin. Se jättää huomioimatta muiden käyttäjien aiheuttamat kuormitukset, yhteyden kytkeytymisen aikana tapahtuvat virheet, itse puhelun aikana tapahtuvat virheet sekä yhteyden purkamisen aikana tapahtuvat virheet. Operaattoreille tärkeät tiedot puhelun kestosta ja käytetyistä palveluista eivät tallennu jatkokäyttöä varten, esimerkiksi laskutustietojen saaminen tallennetusta tietomäärästä on erittäin vaikeata. Käyttäjän ohjelmalta saamat vasteet ovat merkkipohjaisia, mikä osaltaan heikentää toteutetun ratkaisun sopivuutta puheeseen voimakkaasti keskittyvässä ympäristössä. Alla onkin esitetty ratkaisumenetelmiä, joilla simulaattorityökalun toteutusta voidaan parantaa vastaamaan enemmän oikeata älyverkkopalvelun käyttöä.

Massageneraattori

Järjestelmään on luotava puhelinverkossa liikennettä simuloiva automaatti, jonka tehtävänä on kuormittaa palvelun kytkentäpistettä suorittamalla palvelun kytkentäpyyntöjä. Luotavalla generaattorilla tulee olla yleinen rajapinta, joka mahdollistaa käytettävien vasteaikojen satunnaisuuden ja puhelinliikenteen satunnaisuuden asettamisen ja muuttamisen. Kaikista suoritetuista toimenpiteistä täytyy jäädä tiedostoon merkintä, jonka perusteella voidaan määrittää tunnuslukuja järjestelmän suorituskyvylle.

Ratkaisun tulee olla sellainen, että samanaikaisesti voidaan testata useita eri palveluita ilman, että ohjelmisto täytyy kääntää uudestaan. Tämä voidaan toteuttaa luomalla massageneraattorille kuvauskieli, joka määrittää halutut generaattorin parametrit.

Virhetilanteiden generoiminen

Ohjelmistoon täytyy luoda menetelmä, jolla voidaan tuottaa testattavaan protokollaan "satunnaisia" virheitä. Erilaisia satunnaisia virheitä voivat olla esimerkiksi väärien toimintojen luominen, jolloin automaattien vasteet vastaanotettuihin viesteihin ovat virheellisiä, parametrien virheellisyydet, joissa lähetettävää informaatiota tietoisesti vääristetään, koodaukselliset virheet, jotka kuvaavat koodauksen aikana tapahtuvia bittivirheitä sekä sisällölliset virheet, joissa lähetettävät viestit ovat muuten oikeita, mutta niiden sisältämä informaatio on väärää (oikean tunnusnumeron sijasta lähtetäänkin virheellinen tunnus).

Erilaisia virheellisiä tapahtumia voidaan saada aikaan useallakin eri tavalla. Automaatit voidaan luoda sellaisiksi, että tietyissä tilanteissa vastanotettuun viestiin lähetetäänkin vasteeksi vääränlainen viesti, ajastimien laukeaminen voi saada aikaan toivotun virhetilanteen, ohjelmisto voi satunnaisesti muuttaa lähetettävän viestin bittejä tai ohjelmiston alustustiedostoissa tai käyttäjän syötteissä annetaan oikean informaation sijasta virheellistä informaatiota. Toteutuksessa voidaan käyttää edellisessä kohdassa mainittuja ratkaisuja.

Laskutustietojen kerääminen

Palvelun kytkentäkeskukselta saatavat puhelutietueet täytyy kyetä tallentamaan tiedostoihin niiden jatkokäsittelyä varten. Tällä hetkellä esimerkiksi laskutustietojen luominen on mahdotonta, koska ei ole saatavilla tarkkoja tietoja puhelun kulusta.

Äänikortti

Tällä hetkellä järjestelmä on suunniteltu toimivan UNIX®-käyttöjärjestelmän päällä. Erottamalla järjestelmästä puhelinta mallintava osuus voidaan laitteiston käytettävyyttä parantaa huomattavasti siten, että osa käytettyä ohjelmistoa voidaan siirtää äänikortilla varustettuun henkilökohtaiseen tietokoneeseen. Käytetyllä äänikortilla voidaan mallintaa oikeassa puhelussa kuultavat äänimerkit ja -viestit.

Liitännät muihin kytkentäkeskuksiin

Järjestelmällä ei tässä vaiheessa voida mallintaa avustavan kytkentäkeskuksen toimintoja. Kyseisen toiminnon mallintaminen tulee kysymykseen silloin, kun simuloidaan tilannetta, jossa palvelunkytkentäpisteestä puuttuu mahdollisuus välittää viestejä palvelun käyttäjälle sekä vastaanottaa hänen antamiaan vasteita.

7 YHTEENVETO

Älyverkkoteknologian nopea kehittyminen on saanut aikaan sen, että eri standardisoimisjärjestöt ovat olleet pakotettuja tekemään standardeja jatkuvasti kehittyvälle alueelle. Olemassa olevien ja kehitteillä olevien standardien erilaisuus johtuu suurelta osin niiden erilaisista lähtökohdista. Yhdysvaltalaisen Bellcoren taustavaikuttajat ovat saaneet aikaan sen, että Bellcore on lähtenyt liikkeelle määrityksissään käytännölliseltä puolelta, kun taas kansainvälisen ITU-T:n on täytynyt määrityksissään ottaa huomioon standardisoimistyöhön osallistuvien maiden tarpeet.

ITU-T:n ja Bellcoren määrittelemien standardien seuraava kehitysvaihe tulee olemaan molempien järjestöjen standardien yhtenäistäminen. Älyverkkoarkkitehtuurien kehittyminen nykyisestä tilanteesta saa aikaan sen, että tulevaisuudessa uudet älyverkkopalvelut ovat aikaisempaa nopeammin määriteltävissä, testattavissa ja käyttöönotettavissa. Uudet älyverkkopalvelut tulevat olemaan myös aikaisempaa helpommin ylläpidettäviä.

Tämän diplomityön aikana on toteutettu älyverkon kytkentäpisteen simulointityökalu, joka perustuu ITU-T:n CS-1 –suosituksiin. Simulointityökaluun on lisäksi sisällytetty osia Nokia Telecommunications Oy:n määrittelemästä älyverkon sovellusprotokollasta.

Toteutettu järjestelmä on osoittanut, että olemassa olevat standardit eivät vielä ole täydellisiä, vaan ne jättävät joidenkin älyverkkoarkkitehtuurin sisältämien osien välisen viestinnän toteutuksen tekijän määriteltäviksi. Tämä saattaa tulevaisuudessa aiheuttaa ongelmia, ellei puutteellisia kohtia määritellä kansainvälisesti. Simulaattorityökalun toteutus on kuitenkin osoittanut, että kehitystä tällä alueella kannattaa jatkaa.

LÄHDELUETTELO

Ber92 Berman R. K.; Brewster J. H. 1992. *Perspectives on the AIN Architecture.* IEEE Communications Magazine. 30, 2, s. 27-32.

Col92 Collet P.; Kung R. 1992. *The Intelligent Network in France.* IEEE Communications Magazine. 30, 2, s. 82-9.

Dur92 Duran J. M.; Visser J. 1992. *International Standards for Intelligent Networks.* IEEE Communications Magazine 30, 2, s. 34-42.

Hom92 Homa J.; Harris S. 1992. *Intelligent Network Requirements for Personal Communications Services.* IEEE Communications Magazine. 30, 2, s. 70-6.

Itu93a ITU-T Q.1201. 1993. *Principles of Intelligent Network Architecture.* International Telecommunication Union. 33 s.

Itu93b ITU-T Q.1202. 1993. *Intelligent Network - Service Plane Architecture.* International Telecommunication Union. 4 s.

Itu93c ITU-T Q.1203. 1993. *Intelligent Network - Global Functional Plane architecture.* International Telecommunication Union. 12 s.

Itu93d ITU-T Q.1204. 1993. *Intelligent Network Distributed Functional Plane Architecture.* International Telecommunication Union. 24 s.

Itu93e ITU-T Q.1205. 1993. *Intelligent Network Physical Plane Architecture.* International Telecommunication Union. 6 s.

Itu93f ITU-T Q.1211. 1993. *Introduction to Intelligent Network Capability Set 1.* International Telecommunication Union. 30 s.

Itu93g ITU-T Q.1213. 1993. *Global Functional Plane for Intelligent Network CS-1.* International Telecommunication Union. 33 s.

Itu93h ITU-T Q.1214. 1993. *Distributed Functional Plane for Intelligent Network CS-1.* International Telecommunication Union. 109 s.

Itu93i ITU-T Q.1215. 1993. *Physical Plane for Intelligent Network CS-1.* International Telecommunication Union. 8 s.

Joh92 Johnson M. A.; Zebo T. J. 1992. *New service testing functions for advanced intelligent networks.* NOMS '92. Networks Without Bounds, Memphis, TN, U.S.A., 6.-9. 4. 1992. New York. IEEE. Vol. 3 s. 709-20.

Kar93 Karttunen T. 1993. *Keskustelut.* 23.8.1993. Telekehityskeskus.

Kun92 Kung R. 1992. *Rationale for Intelligent Networks.* IFIP transactions C-1. s. 69-79.

Lew81 Lewis H. R.; Papadimitriou C.H. 1981. *Elements of the theory of computation.* Englewood Cliffs, NJ, U.S.A. Prentice-Hall Inc. 466 s.

Li93 Li M.; Nichols E. 1993. *Intelligent Networks: Strategies for Customized Global Services*. London, England. Ovum Ltd. 315 s.

Mal93 Malka J.; Ojanperä E. 1993. *User's Guide for CVOPS 4.0*. Espoo. Technical Research Centre of Finland. 102 s.

Nat92 Natarajan N.; Slawsky G. M. 1992. *A Framework Architecture for Multimedia Information Networks*. IEEE Communications Magazine. 30, 2, s. 97-104.

Sch92 Schulz K.; Kaufer H.; Heilig B. 1992. *The Deutsche Bundespost Telekom Implements IN*. IEEE Communications Magazine. 30, 2, s. 90-6.

LIITTEET

A Automaatit

A.1 Phone.aut

```
/*
(*************************************************************
 *
 *    DATE:                31.08.1994
 *
 *    AUTHORS:             Ahti Rissanen
 *
 *    MODULE NAME:         PHONE.AUT
 *
 *************************************************************)

(* State    Input              Actions *)

Idle        Lift        {
      if(0 == prompt())
            to(Idle)
      else
            {
                  DOWN.Phone_Digits_Collected
                  to(Ph_Off_Hook)
            }
}

Ph_Off_Hook    Drop       {
      DOWN.Receiver_On_Hook
      to(Idle)
}

Ph_Off_Hook    stor_prompt      {
      if(0 == prompt())
            {
                  DOWN.Receiver_On_Hook
                  to(Idle)
            }
      Phone_Digits
}

Ph_Off_Hook    Feature_Req      {
      Feature
}

*/
```

A.2 Ccaf_a.aut

```
/*
(*************************************************************
 *
 *   DATE:              31.08.1994
 *
 *   AUTHORS:           Ahti Rissanen
 *
 *   MODULE NAME:       CCAF_A:AUT
 *
 *************************************************************)

(* State    Input              Actions *)

Global          Other           {}

Anystate  UP.Subsequent_Address     {
     Ccaf_Subsequent_Address
}

Anystate  Ssp_ToAnalyzing           {
     to(Analyzing_Information)
}

Anystate  Ssp_ToCollecting     {
     to(Collecting_Information)
}

Anystate  UP.Receiver_On_Hook     {
     stop(all)
     if(conn == 1)
          O_Disconnect
     Ccaf_Clear_Forward
     to(O_NullAndOriginatingAttemptAuthorized)
}

Anystate  UP.feature      {
     stop(all)
     ccaf_Feature_Request
}

Anystate  Ssp_Congestion        {
     to(O_NullAndOriginatingAttemptAuthorized)
}

O_NullAndOriginatingAttemptAuthorized  Phone_Digits_Collected    {
     conn = 0
     empty = 0
     enter_cli()
     Check_Trigger
     to(Analyzing_Information)
}

Collecting_Information    Dial        {
     enter_cli()
     Check_Trigger
     to(Analyzing_Information)
}

Analyzing_Information     Ssp_Address_Complete      {
     to(Analyzing_Information)
}

Analyzing_Information     UP.Subsequent_Address     {
     Ccaf_Subsequent_Address
}

Analyzing_information     Ssp_Forced_Release   {
     to(O_NullAndOriginatingAttemptAuthorized)
}

Analyzing_Information     Ssp_Call_Failure      {
     to(O_NullAndOriginatingAttemptAuthorized)
}

Analyzing_Information     Ssp_Answer          {}

Analyzing_Information     Ssp_New_Call        {
     conn = 1
     ADDRESS
     start(Wf_Address_Complete_Tim)
     to(Routing_AndAlerting)
}

Analyzing_Information     Ssp_Normal_Call     {
     if(empty == 0)
          {
               conn = 1
               ADDRESS
               start(Wf_Address_Complete_Tim)
               to(Routing_AndAlerting)
          }
     else
          {
               to(O_Active)
          }
}

Routing_AndAlerting       TAnswer   {
     to(O_Active)
}

O_Active        CLEAR_BACK        {
     to(O_NullAndOriginatingAttemptAuthorized)
}

*/
```

A.3 Ccaf_b.aut

```
/*
(*************************************************************
 *
 *   DATE:              31.08.1994
 *
 *   AUTHORS:           Ahti Rissanen
 *
 *   MODULE NAME:       CCAF_B.AUT
 *
 *************************************************************)

(* Mgmt interface *)
(* Force B -subscriber busy *)

T_NullAndAuthorizeTerminationAttempt   ToBusy    {
      resp = 0
      to(T_Active)
}

T_Active          ToIdle     {
      resp = 1
      to(T_NullAndAuthorizeTerminationAttempt)
}

T_NullAndAuthorizeTerminationAttempt   ADDRESS        {
      if(1 == testbusy())
            {
                  SUBSCRIBER_BUSY_SIGNAL
                  delete(recframe)
            }
      else
            {
                  ADDRESS_COMPLETE
                  delete(recframe)
                  to(T_Alerting)
            }
}

T_Alerting        answer     {
      TAnswer
      resp = 1
      to(T_Active)
}

T_Alerting        O_Disconnect    {
      to(T_NullAndAuthorizeTerminationAttempt)
}

T_Active          ADDRESS         {
      SUBSCRIBER_BUSY_SIGNAL
      delete(recframe)
}

T_Active          On_Hook    {
      if(resp == 1)
            {
                  CLEAR_BACK
                  to(T_NullAndAuthorizeTerminationAttempt)
            }
}

T_Active          O_Disconnect    {
      to(T_NullAndAuthorizeTerminationAttempt)
}

*/
```

A.4 Storserv.aut

```
/*
(*************************************************************
 *
 *   DATE:              31.08.1994
 *
 *   AUTHORS:           Ahti Rissanen
 *
 *   MODULE NAME:       STORSERV.AUT
 *
 *************************************************************)

(* State    Input       Actions *)

global      other       {}

Idle            Ssp_Announcement    {
      field_nr = 0
      show_announcement()
      get_field_count()
      if(fields > 0)
            {
                  if(phone_CEP == 0)
                        {
                              while(fields > 0)
                                    {
                                          inp_field();
                                          --fields;
                                          save_digits();
                                          ++field_nr
                                    }
                              Send_To_Resource_Complete
                              to(Idle)
                        }
                  else
                        {
                              start(T1)
                              Stor_Prompt
                              to(prompting)
                        }
            }
      else
            to(Idle)
}

Prompting       Timeout(T1)     {
      to(Idle)
}

Prompting       Phone_Digits    {
      stop(T1)
      --fields
      if(0 == save_digits())
            to(Idle)
      else
            {
            if(0 == fields)
                  {
                        Send_To_Resource_Complete
                        to(Idle)
                  }
            else
                  {
                        ++field_nr
                        Stor_Prompt
                  }
            }
}*/
```

A.5 Ssp.aut

```
/*
(*************************************************************
 *
 *   DATE:              31.08.1994
 *
 *   AUTHORS:           Ahti Rissanen
 *
 *   MODULE NAME:       SSP.AUT
 *
 *************************************************************)

(* Useful macros *)

void        zero_event_mask()    {
      call_delivered_event = 0
      answered_event = 0
      cleared_event = 0
      feature_requested_event = 0
      disconnected_event = 0
      release_timeout_event = 0
      invalid_info_event = 0
      network_busy_event = 0
      busy_reported_event = 0
      resource_clear_event = 0
      timeout_occurred_event = 0
      route_failure_event = 0
}

void        t001()      {
      analyze_digits()
      trigger_1_check_point()
      if(trigger == 1)
            {
                  Communication_type = 2
                  AnalyzedInformation
                  start(WF_Instructions_timer)
                  Tc_Begin_Request
                  to(WF_Instructions)
            }
      else
            {
                  Ssp_Normal_Call
                  normal_call_handling()
                  zero_event_mask()
                  to(Idle)
            }
}

void        send_response()         {
      Tc_Continue_Request
      start(WF_Instructions_timer)
      to(WF_Instructions)
}

(* State    Input       Actions *)

Anystate    O_Disconnect    {
      if(AbandonDP == 0)
            {
                  if(Call_Information_Request_Event == 0)
                        to(Idle)
                  else
                        {
                              Call_Information_Request()
                              to(Idle)
                        }
            }
      else
            {
                  if(AbandonDP == 1)
                        {
                              Event_Report_BCSM()
                              to(WF_Instructions)
                        }
                  if(AbandonDP == 2)
                        {
                              if(Call_Information_Request_Event == 0)
                                    {
                                          Event_Report_BCSM()
                                          to(Idle)
                                    }
                              else
                                    {
                                          Event_Report_BCSM()
                                          Call_Information_Report()
                                          to(Idle)
                                    }
                        }
            }
}

Anystate    T_Disconnect    {
      if(DisconnectDP == 0)
            {
                  if(Call_Information_Request_Event == 0)
                        {}
            }
}

Anystate    Tc_Begin_Indication         {}

Anystate    Tc_Continue_Indication      {}

Anystate    Tc_End_Indication     {
      if(too_tired > 0)
      dialog_end = 1
}

global      other       {}

Idle        Ccaf_Address_Complete     {}

Idle        Ccaf_Clear_Forward    {}
```

```
Idle        Ccaf_Subscriber_Answer    {}

Idle        ccaf_congestion           {}

Idle        InitiateCall    {
      start(WF_Instructions_Timer)
      to(WF_Instructions)
}

Idle        Check_Trigger       {
      if(Call_Gapping_event == 1)
            {}
      else
            {
                  if(Service_Blocking_Event == 1)
                        {}
                  else
                        {
                              errorcode = 0
                              thru_connected = 0
                              send_answer = 1
                              Origin_On_Hook = 0
                              t001()
                        }
            }
}

WF_Instructions       Tc_Continue_Indication          {}

WF_Instructions       Tc_End_Indication    {}

WF_Instructions       Ccaf_Clear_Forward   {
      ++Aonhook
}

WF_Instructions (Ccaf_Release_Guard, Ccaf_Clear_Back)
      {++Bonhook}

WF_Instructions       HoldCallInNetwork    {
      Hold_Call_In_Network()
}

WF_Instructions       RequestReportBCSMEvent       {
      Request_Report_BCSM_Event()
}

WF_Instructions       RequestNotificationChargingEvent    {
      Request_Notification_Charging_Event()
}

WF_Instructions       ResetTimer     {
      Reset_Timer()
}

WF_Instructions       FurnishChargingInformation      {
      Furnish_Charging_Information()
}

WF_Instructions       ApplyCharging   {
      Apply_Charging()
}

WF_Instructions CallInformationRequest    {
      stop(WF_Instructions_Timer)
      Call_Information_Request()
}

WF_Instructions       SendChargingInformation   {
      stop(WF_Instructions_Timer)
      Send_Charging_Information()
}

WF_Instructions       ConnectToResource   {
      stop(WF_Instructions_Timer)
      Connect_To_Resource()
      to(WFEO_UserInteraction)
}

WF_Instructions       EstablishTemporaryConnection   {
      stop(WF_Instructions_Timer)
      Establish_Temporary_Connection()
      to(WFEO_TemporaryConnection)
}

WF_Instructions       Connect    {
      stop(WF_Instructions_Timer)
      if((EDP_Event == 1) || (Call_Information_Report_event == 1))
            {
                  Connect()
                  to(Monitoring)
            }
      else
            to(Idle)
}

WF_Instructions       CollectInformation    {
      stop(WF_Instructions_Timer)
      if(Aonhook > 0)
            {
                  stop(all)
                  ODisconnect
                  send_response()
            }
      else
            {
                  if(thru_connected == 0)
                        Ssp_Address_Complete
                  if(send_answer == 1)
                        Ssp_Answer
                  Ssp_Announcement
                  thru_connected = 1
                  send_answer = 0
                  start(WFEO_UserInteraction_Timer)
                  to(WFEO_UserInteraction)
            }
}
```

```
WF_Instructions      AnalyzeInformation  {
     stop(WF_Instructions_Timer)
     if(Aonhook > 0)
          {
               send_response()
          }
     else
          {
               stop(all)
               Origin_On_Hook = 1
               Aonhook = 0
               if((EDP_Event == 1) || (Call_Information_Report_Event == 1))
                    {
                         Analyze_Information()
                         to(Monitoring)
                    }
               else
                    {
                         bcsm_state = 5
                         Ssp_ToAnalyzing
                         t001()
                    }
          }
}

WF_Instructions      SelectRoute     {
     stop(WF_Instructions_Timer)
     if((EDP_Event == 1) || (Call_Information_Report_Event == 1))
          {
               Select_Route()
               to(Monitoring)
          }
     else
          to(Idle)
}

WF_Instructions      SelectFacility   {
     stop(WF_Instructions_Timer)
     if((EDP_Event == 1) || (Call_Information_Report_Event == 1))
          {
               Select_Facility()
               to(Monitoring)
          }
     else
          to(Idle)
}

WF_Instructions      Continue   {
     stop(WF_Instructions_Timer)
     if((EDP_Event == 1) || (Call_Information_Report_Event == 1))
          {
               Continue()
               to(Monitoring)
          }
     else
          to(Idle)
}

WF_Instructions      ReleaseCall     {
     stop(WF_Instructions_Timer)
     if((EDP_Event == 1) || (Call_Information_Report_Event == 1))
          {
               Release_Call()
               to(Monitoring)
          }
     else
          to(Idle)
}

WF_Instructions      Timeout(WF_Instructions_Timer)      {
     Ssp_Call_Failure
     normal_call_handling()
     zero_event_mask()
     to(Idle)
}

WF_Instructions      (Tc_U_Abort_Indication, Tc_P_Abort_Indication)      {
     stop(WF_Instructions_Timer)
     update_abort_counter()
     Ssp_Congestion
     normal_call_handling()
     zero_event_mask()
     to(Idle)
}

WF_Instructions      Tc_U_Error_Indication      {
     stop(WF_Instructions_Timer)
     update_errortype_counter()
     Ssp_Call_Failure
     normal_call_handling()
     zero_event_mask()
     to(Idle)
}

WF_Instructions      other      {}

WFEO_UserInteraction      PlayAnnouncement    {
     start(WFEO_UserInteraction_Timer)
}

WFEO_UserInteraction      PromptAndCollectUserInformation      {
     start(WFEO_UserInteraction_Timer)
}

WFEO_UserInteraction      SpecializedResourceReport      {
     start(WFEO_UserInteraction_Timer)
}


WFEO_UserInteraction      Cancel     {
     start(WFEO_UserInteraction_Timer)
}
WFEO_UserInteraction      SrfDisconnect     {
     start(WF_Instructions_Timer)
     to(WF_Instructions)
}
```

```
WFEO_UserInteraction    Timeout(WFEO_UserInteraction_Timer)    {
    to(Idle)
}

WFEO_UserInteraction    RequestReportBCSMEvent    {
    Request_Report_BCSM_Event()
}

WFEO_UserInteraction    RequestNotificationChargingEvent    {
    Request_Notification_Charging_Event()
}

WFEO_UserInteraction    ResetTimer    {
    Reset_Timer()
}

WFEO_UserInteraction    FurnishChargingInformation    {
    Furnish_Charging_Information()
}

WFEO_UserInteraction    ApplyCharging    {
    Apply_Charging()
}

WFEO_UserInteraction    CallInformationRequest    {
    Call_Information_Request()
}

WFEO_UserInteraction    SendChargingInformation    {
    Send_Charging_Information()
}

WFEO_UserInteraction    DisconnectForwardConnection    {
    start(WF_Instructions_Timer)
    to(WF_Instructions)
}

WFEO_UserInteraction    SrfCallDisconnected    {
    start(WF_Instructions_Timer)
    to(WF_Instructions)
}

WFEO_UserInteraction    Send_To_Resource_Complete    {
    stop(WFEO_UserInteraction_Timer)
    if(response_required == 1)
        {
            ResourceClear
            Tc_Continue_Request
            start(WF_Instructions_Timer)
            to(WF_Instructions)
        }
    else
        {
            zero_event_mask()
            to(Idle)
        }
}

WFEO_UserInteraction    other    {}

WFEO_TemporaryConnection    DisconnectForwardConnection    {
    stop(all)
    start(WF_Instructions_Timer)
    to(WF_Instructions)
}
WFEO_TemporaryConnection    Timeout(WFEO_TemporaryConnection_Timer)    {
    stop(all)
    to(Idle)
}

WFEO_TemporaryConnection    O_Disconnect    {
    stop(all)
    start(WF_Instructions_Timer)
    to(WF_Instructions)
}

WFEO_TemporaryConnection    RequestReportBCSMEvent    {
    Request_Report_BCSM_Event()
}

WFEO_TemporaryConnection    RequestNotificationChargingEvent    {
    Request_Notification_Charging_Event()
}

WFEO_TemporaryConnection    ResetTimer    {
    Reset_Timer()
}

WFEO_TemporaryConnection    FurnishChargingInformation    {
    Furnish_Charging_Information()
}

WFEO_TemporaryConnection    ApplyCharging    {
    Apply_Charging()
}

WFEO_TemporaryConnection    CallInformation    {
    Call_Information()
}

WFEO_TemporaryConnection    SrfDisconnect    {
    stop(all)
    start(WF_Instructions_Timer)
    to(WF_Instructions)
}

WFEO_TemporaryConnection    other    {}

Monitoring    RequestReportBCSMEvent    {
    Request_Report_BCSM_Event()
}

Monitoring    RequestNotificationChargingEvent    {
    Request_Notification_Charging_Event()
}
```

```
Monitoring      CallInformationRequest    {
    Call_Information_Request()
}

Monitoring      ReleaseCall     {
    if(Call_Information_request_event == 1)
        Call_Information_Report()
    Release_Call()
    to(Idle)
}

Monitoring      other       {}

Monitoring      Ccaf_Address_Complete     {
    stop(all)
    if(Call_Delivered_Event >= 10)
        {
            if((Call_Delivered_Event - 10) > 0)
                {
                    send_response()
                }
        }
}

Monitoring      Ccaf_Subscriber_Busy      {
    stop(all)
    if(Busy_Reported_Event >= 10)
        {
        if(Busy_Reported_Event == 11)
            {
                send_response()
            }
        }
}

Monitoring      Ccaf_Congestion       {
    stop(all)
    if(Network_Busy_Event >= 10)
        {
            if(Network_Busy_Event == 11)
                {
                    RouteSelectFailure
                    send_response()
                }
        }
}

Monitoring      Ccaf_Clear_Forward  {
    stop(all)
    Origin_On_Hook = 1
    if(Disconnected_Event >= 10)
        {
            if(Disconnected_Event == 11)
                {
                    ODisconnect
                    send_response()
                }
        }
}

Monitoring      Ccaf_Invalid_Destination  {
    stop(all)
    if(Invalid_Info_Event >= 10)
        {
            if(Invalid_Info_Event == 11)
                {
                    send_response()
                }
        }
}

Monitoring      Ccaf_Subscriber_Answer    {
    stop(all)
    if(Answered_Event >= 10)
        {
            Feature_Requested_Event = 0
            if(Answered_Event == 11)
                {
                    send_response()
                }
        }
}

Monitoring (Ccaf_Release_Guard, Ccaf_Clear_Back)
{
    stop(all)
    if(Cleared_Event >= 10)
        {
            if(Cleared_Event == 11)
                {
                    send_response()
                }
        }
}

Monitoring      Ccaf_Feature_Request      {
    stop(all)
    if(Feature_Requested_Event >= 10)
        {
            if(Feature_Requested_Event == 11)
                {
                    send_response()
                }
        }

*/
```

B Ssp:n sisäiset muuttujat

```
/*****************************************************************
 *
 *   DATE:              31.08.1994
 *
 *   AUTHORS:           Ahti Rissanen
 *
 *   MODULE NAME:       SSP.IN
 *
 *****************************************************************/

/****************
 *
$    Variables
 *
 ****************/

#define   AbandonDP                          0
#define   Active_Dialogue                    1
#define   Aonhook                            2
#define   Bonhook                            3
#define   Call_Gapping_Event                 4
#define   Call_Information_Report_Event      5
#define   Call_Information_Request_Event     6
#define   Communication_Type                 7
#define   Dialogue_Id                        8
#define   DisconnectDP                       9
#define   EDP_Event                          10
#define   Event_Mask                         11
#define   Event_Parameter                    12
#define   Service_Blocking_Event             13
#define   Answered_Event                     14
#define   Bcsm_State                         15
#define   Busy_Reported_Event                16
#define   Call_Delivered_Event               17
#define   Cleared_Event                      18
#define   Dialog_End                         19
#define   Disc_Case                          20
#define   Disconnected_Event                 21
#define   Error_Code                         22
#define   Feature_Requested_Event            23
#define   First_Ann                          24
#define   Invalid_Info_Event                 25
#define   Max_Call_Time                      26
#define   Network_Busy_Event                 27
#define   Operation_Code                     28
#define   Origin_On_Hook                     29
#define   Phone_CEP                          30
#define   Release_Timeout_Event              31
#define   Resource_Clear_Event               32
#define   Response_Required                  33
#define   Route_Failure_Event                34
#define   Second_Ann                         35
#define   Send_Answer                        36
#define   Thru_Connected                     37
#define   Timeout_Occurred_Event             38
#define   Too_Tired                          39
#define   Trigger                            40

/****************
 *
$    States
 *
 ****************/

#define   Idle                               0
#define   Trigger_Processing                 1
#define   WF_Instructions                    2
#define   WFEO_UserInteraction               3
#define   WFEO_TemporaryConnection           4
#define   Monitoring                         5

/****************
 *
$    Pdus
 *
 ****************/

#define   ActivityTest                       0
#define   AnalyzeInformation                 1
#define   AnalyzedInformation                2
#define   ApplyCharging                      3
#define   ApplyChargingReport                4
#define   AssistChargingReport               5
#define   AssistRequestInstructions          6
#define   CallGap                            7
#define   CallInformation                    8
#define   CallInformationReport              9
#define   CallInformationRequest             10
#define   Cancel                             11
#define   CancelResource                     12
#define   CancelStatusReport                 13
#define   CollectInformation                 14
#define   CollectedInformation               15
#define   Connect                            16
#define   ConnectToResource                  17
#define   Continue                           18
#define   DisconnectForwardConnection        19
#define   EstablishTemporaryConnection       20
#define   EventNotificationCharging          21
#define   EventReportBCSM                    22
#define   FurnishChargingInformation         23
#define   HoldCallInNetwork                  24
#define   InitialDP                          25
#define   InitiateCall                       26
#define   InitiateCallAttempt                27
#define   OAnswer                            28
#define   OCalledPartyBusy                   29
#define   ODisconnect                        30
#define   OMidCall                           31
#define   ONoAnswer                          32
#define   O_Disconnect                       33
#define   OriginationAttemptAuthorized       34
```

```
#define    PlayAnnouncement                    35
#define    PromptAndCollectUserInformation     36
#define    ReleaseCall                         37
#define    RequestCurrentStatusReport          38
#define    RequestEveryStatusChangeReport      39
#define    RequestFirstStatusMatchReport       40
#define    RequestNotificationChargingEvent    41
#define    RequestReportBCSMEvent              42
#define    ResetTimer                          43
#define    ResourceClear                       44
#define    RouteSelectFailure                  45
#define    SelectFacility                      46
#define    SelectRoute                         47
#define    SendChargingInformation             48
#define    ServiceFilteringResponse            49
#define    SpecializedResourceReport           50
#define    SrfCallDisconnected                 51
#define    SrfDisconnect                       52
#define    StatusReport                        53
#define    TAnswer                             54
#define    TCalledPartyBusy                    55
#define    TDisconnect                         56
#define    TMidCall                            57
#define    TNoAnswer                           58
#define    TermAttemptAuthorized               59

/*****************
 *
$    Procedures
 *
 ****************/

#define    analyze_digits()                    0
#define    analyze_information()               1
#define    apply_charging()                    2
#define    call_information()                  3
#define    call_information_report()           4
#define    call_information_request()          5
#define    collect_information()               6
#define    connect()                           7
#define    connect_to_resource()               8
#define    continue()                          9
#define    create_call_detail_record()         10
#define    establish_temporary_connection()    11
#define    event_report_bcsm()                 12
#define    furnish_charging_information()      13
#define    hold_call_in_network()              14
#define    normal_call_handling()              15
#define    play_announcement_to_caller()       16
#define    release_call()                      17
#define    request_notification_charging_Event()  18
#define    request_report_bcsm_Event()         19
#define    reset_timer()                       20
#define    select_facility()                   21
#define    select_route()                      22
#define    send_charging_information()         23
#define    trigger_1_check_point()             24
#define    trigger_2_check_point()             25
#define    update_abort_counter()              26
#define    update_errortype_counter()          27

/*****************
 *
$    Timers
 *
 ****************/

#define    WF_Instructions_Timer               0
#define    WFEO_UserInteraction_Timer          2
#define    WFEO_TemporaryConnection_Timer      3
#define    Monitoring_Timer                    4

/*****************
 *
$    timer tick counts
 *
 *****************

WF_Instructions_Timer                          60
WFEO_UserInteraction_Timer                     60
WFEO_TemporaryConnection_Timer                 60
Monitoring_Timer                               60

*/

/*****************
 *
$    End
 *
 *****************/

typedef struct {
      PARAMETERS peerparameters;
      char u_digits[80];
      char datadigits[80];
      char u_cli[80];
      STORBLOCK storblock;
      RESULTFIELDS resultfields;
} SSP_REF;
```

C Ssp:n ulkoiset viestit

```
/*************************************************************
 *
 *   DATE:              31.08.1994
 *
 *   AUTHORS:           Ahti Rissanen
 *
 *   MODULE NAME:       SSP.IF
 *
 *************************************************************/

/****************
 *
$    Primitives
 *
 ****************/

#define    Check_Trigger                0
#define    Ccaf_Subsequent_Address      1
#define    Ccaf_Answer                  2
#define    Ccaf_Address_Complete        3
#define    Ccaf_Clear_Forward           4
#define    Ccaf_Subscriber_Busy         5
#define    Ccaf_Release_Guard           6
#define    Ccaf_Clear_Back              7
#define    Ssp_New_Call                 8
#define    Ssp_Answer                   9
#define    Ssp_Congestion               10
#define    Ssp_Clear_Forward            11
#define    Ssp_Call_Failure             12
#define    Ssp_Forced_Release           13
#define    Ssp_ToAnalyzing              14
#define    Ssp_Address_Complete         15
#define    Ssp_ToCollecting             16
#define    Ccaf_Feature_Request         17
#define    Ccaf_Congestion              18
#define    Ccaf_Subscriber_Answer       19
#define    Ccaf_Timeout                 20
#define    Ccaf_Relt_Out                21
#define    Ccaf_Invalid_Destination     22
#define    Ssp_Normal_Call              23
#define    T_Disconnect                 24

/****************
 *
$    End
 *
 ****************/

typedef struct {
      char digits[80];
      char cli[80];
      INT32 ph_CEP;
} SSPS_PARAM_BLOCK;
```

D Ssp:n ohjelmakoodi

D.1 Ssp.c

```
/*****************************************************************
 *
 *   DATE:              31.08.1994
 *
 *   AUTHORS:           Ahti Rissanen
 *
 *   MODULE NAME:       SSP.C
 *
 *****************************************************************/

#include  "enviro.def"
#include  "general.def"
#include  "exttypes.typ"
#include  "numb.def"
#include  "cod.def"
#include  "clibrary.ext"
#include  "runauto.ext"
#include  "messages.ext"
#include  "timers.ext"
#include  "utils.ext"
#include  "memory.ext"
#include  "frame.ext"
#include  "connects.ext"

#include  "ssp.if"
#include  "tc.if"
#include  "sspstor.if"

#include  "ssp.in"
#include  "ssp.aut"

#include  "sspcod.c"

/*****************************************************************
 *
 *   STATIC FUNCTION:       puttcSrvParameters
 *
 *****************************************************************/

static void puttcSrvParameters (msg)
     MESSAGE *msg;
{
     VTASK *vtask = msg -> source;
     INT32 *var = vtask -> var;
     SSP_REF *ref = (SSP_REF *) vtask -> ref;
     char el[132];
     TCS_PARAM_BLOCK *block = (TCS_PARAM_BLOCK *) wzalloc(sizeof(TCS_PARAM_BLOCK));
     PARAMETERS help;

     switch (msg -> primitive.item)
          {
               case tc_invoke_request:
               case tc_result_l_request:
                    block -> operation = (INT16) var[Operation_Code];
                    block -> dialogue_id = var[Dialogue_Id];
                    break;

               case Tc_Begin_Request:
               case Tc_Continue_Request:
               case tc_end_request :
               case tc_uni_request :
               case tc_u_reject_request:
               case tc_u_error_request:
               case tc_u_abort_request:
                    block -> dialogue_id = var[Dialogue_Id];
                    break;

               default:
                    sprintf(el, "\n*** Unknown primitive: ");
                    getSymbol(&msg -> primitive, vtask -> type -> directory, el+strlen(el));
                    sprintf(el+strlen(el), ", check SOURCE -list");
                    traceLine(el);
                    break;
          }

     help = ref -> peerparameters;
     (*block) .parameters = help;
     msg -> parameters = (BYTE_TYPE *) block;
}

/*****************************************************************
 *
 *   STATIC FUNCTION:       gettcSrvParameters
 *
 *****************************************************************/

static void gettcSrvParameters (msg)
     MESSAGE *msg;
{
     VTASK *vtask = msg -> dest;
     INT32 *var = vtask -> var;
     char el[132];
     TCS_PARAM_BLOCK *block = (TCS_PARAM_BLOCK *) msg -> parameters;
     PARAMETERS help;
     SSP_REF *ref = (SSP_REF *) vtask -> ref;

     switch (msg -> primitive.item)
          {
               case tc_invoke_indication:
               case tc_result_l_indication:
                    var[Dialogue_Id] = block -> dialogue_id;
                    var[Operation_Code] = (INT32) block -> operation;
                    help = (*block) .parameters;
                    ref -> peerparameters = help;
                    msg -> primitive.item = var[Operation_Code];
                    sspDecode(vtask);
                    msg -> primitive.selector = PEER;
                    break;
```

```
            case Tc_Begin_Indication:
            case Tc_Continue_Indication:
            case Tc_End_Indication:
            case tc_uni_indication:
            case tc_u_reject_indication:
            case tc_l_reject_indication:
            case Tc_U_Error_Indication:
            case Tc_U_Abort_Indication:
                var[Dialogue_Id] = block -> dialogue_id;
                break;

            default:
                sprintf(el, "\n*** Unknown primitive:          ");
                getSymbol(&msg -> primitive, vtask -> type -> directory, el+strlen(el));
                sprintf(el+strlen(el), ", check DEST -list\n");
                traceLine(el);
                break;
        }
}

/*************************************************************
 *
 *    STATIC FUNCTION:        putsspSrvParameters
 *
 *************************************************************/

static void putsspSrvParameters (msg)
    MESSAGE *msg;
{
    VTASK *vtask = msg -> source;
    SSPS_PARAM_BLOCK *block = (SSPS_PARAM_BLOCK *)
    wzalloc(sizeof(SSPS_PARAM_BLOCK));
    SSP_REF *ref = (SSP_REF *) vtask -> ref;

    switch (msg -> primitive.item)
        {
            case Ssp_Address_Complete:
                break;

            case Ssp_New_Call:
            case Ssp_Normal_Call:
                strcpy(block -> digits, ref -> peerparameters.collected_info);
                break;

            default:
                break;
        }

    msg -> parameters = (BYTE_TYPE *) block;
}

/*************************************************************
 *
 *    STATIC FUNCTION:        putsspstorParameters
 *
 *************************************************************/

static void putsspstorParameters (msg)
    MESSAGE *msg;
{
    VTASK *vtask = msg -> source;
    INT32 *var = vtask -> var;
    SSPSTOR_PARAM_BLOCK *block = (SSPSTOR_PARAM_BLOCK *)
    wzalloc(sizeof(SSPSTOR_PARAM_BLOCK));
    SSP_REF *ref = (SSP_REF *) vtask -> ref;
    int i;

    switch (msg -> primitive.item)
        {
            case Ssp_Announcement:
                for(i = 0; i < 4;i++)
                    {
                        block -> storblock[i].ann = ref -> storblock[i].ann;
                        block -> storblock[i].interruptable = ref -> storblock[i].interruptable;
                        block -> storblock[i].digits = ref -> storblock[i].digits;
                        block -> storblock[i].len = ref -> storblock[i].len;
                    }
                block -> ph_CEP = var[Phone_CEP];
                break;

            default:
                traceLine("Unknown primitive");
                break;
        }

    msg -> parameters = (BYTE_TYPE *) block;
}

/*************************************************************
 *
 *    STATIC FUNCTION:        getsspstorParameters
 *
 *************************************************************/

static void getsspstorParameters (msg)
    MESSAGE *msg;
{
    VTASK *vtask = msg -> dest;
    SSP_REF *ref = (SSP_REF *) vtask -> ref;
    SSPSTOR_PARAM_BLOCK *block = (SSPSTOR_PARAM_BLOCK *) msg -> parameters;
    int i;

    switch(msg -> primitive.item)
        {
            case Send_To_Resource_Complete:
                for(i = 0; i < 4;i++)
                    {
                        strcpy(ref -> resultfields[i], block -> resultfields[i]);
                    }
                break;

            default:
                traceLine("Unknown primitive\n");
                break;
        }
}
```

```
/*****************************************************************
 *
 *    STATIC FUNCTION:        getsspSrvParameters
 *
 *****************************************************************/

static void getsspSrvParameters (msg)
     MESSAGE *msg;
{
     VTASK *vtask = msg -> dest;
     INT32 *var = vtask -> var;
     SSP_REF *ref = (SSP_REF *) vtask -> ref;
     INT16 item = msg -> primitive.item;
     SSPS_PARAM_BLOCK *block = (SSPS_PARAM_BLOCK *) msg -> parameters;
     char *p, *strchr(), *digits = block -> digits;

     switch (item)
          {
               case Check_Trigger:
                    strcpy(ref -> u_digits, digits);
                    strcpy(ref -> datadigits, digits);
                    strcpy(ref -> u_cli, block -> cli);
                    var[Phone_CEP] = block -> ph_CEP;
                    break;

               case Ccaf_Subsequent_Address:
                    strcpy(ref -> u_digits, digits);
                    strcat(ref -> datadigits, ref -> u_digits);
                    break;

               default:
                    break;
          }
}

/*****************************************************************
 *
 *    STATIC FUNCTION:        freesspSrvParameters
 *
 *****************************************************************/

static void freesspSrvParameters(msg)
     MESSAGE *msg;
{
     SSPS_PARAM_BLOCK *block = (SSPS_PARAM_BLOCK *) msg -> parameters;

     if(block)
          wfree(block);
}

/*****************************************************************
 *
 *    STATIC FUNCTION:        putsspPeerParameters
 *
 *****************************************************************/

static void putsspPeerParameters(msg)
     MESSAGE *msg;
{
     VTASK *vtask = msg -> source;
     uint16 pdu = msg -> primitive.item;
     SSP_REF *ref = (SSP_REF *) vtask -> ref;

     switch (pdu)
          {
               case AnalyzedInformation:
               case AnalyzeInformation:
               case PromptAndCollectUserInformation:
               case CancelResource:
               case Continue:
               case O_Disconnect:
               case CallInformationRequest:
               case InitiateCall:
               case HoldCallInNetwork:
               case RequestReportBCSMEvent:
               case RequestNotificationChargingEvent:
               case ResetTimer:
               case FurnishChargingInformation:
               case ApplyCharging:
               case SendChargingInformation:
               case ConnectToResource:
               case Connect:
               case EstablishTemporaryConnection:
               case SelectRoute:
               case SelectFacility:
               case ReleaseCall:
               case PlayAnnouncement:
               case SpecializedResourceReport:
               case Cancel:
               case SrfDisconnect:
               case SrfCallDisconnected:
               case CallInformation:
               case DisconnectForwardConnection:
                    reset_peerparameters(&(ref -> peerparameters));
                    sspEncode(pdu, vtask);
                    vtask -> var[Operation_Code] = pdu;
                    msg -> primitive.item = tc_invoke_request;
                    break;

               case ODisconnect:
               case RouteSelectFailure:
               case ResourceClear:
               case CollectInformation:
                    reset_peerparameters(&(ref -> peerparameters));
                    sspEncode(pdu, vtask);
                    vtask -> var[Operation_Code] = pdu;
                    msg -> primitive.item = tc_result_l_request;
                    break;

               default:
                    traceLine("\n\nInvalid PDU!!!\n");
                    break;
          }

     msg -> primitive.selector = DOWN;
     puttcSrvParameters(msg);
}
```

```
/*****************************************************************
 *
 *    STATIC FUNCTION:        analyze_digits
 *
 *****************************************************************/

INT32 Analyze_digits (vtask)
      VTASK *vtask;
{
      SSP_REF *ref = (SSP_REF *) vtask -> ref;
      char *tmp = ref -> datadigits;

      if(!strncmp("9200", tmp, 4) || !strncmp("9700", tmp, 4)
      || !strncmp("9800", tmp, 4) || !strncmp("9850", tmp, 4)
      || !strncmp("9300", tmp, 4) || !strncmp("9350", tmp, 4))
            {
                  vtask -> var[Trigger] = 1;
                  strcpy(ref -> u_digits, tmp);
            }
      else
            {
                  vtask -> var[Trigger] = 0;
                  strcpy(ref -> peerparameters.collected_info, tmp);
      }
}

/*****************************************************************
 *
 *    STATIC FUNCTION:        normal_call_handling
 *
 *****************************************************************/

static INT32 Normal_call_handling (vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        create_call_detail_record
 *
 *****************************************************************/

static INT32 Create_call_detail_record (vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        play_announcement_to_caller
 *
 *****************************************************************/

static INT32 Play_announcement_to_caller (vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        update_abort_counter
 *
 *****************************************************************/

static INT32 Update_abort_counter(vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        update_errortype_counter
 *
 *****************************************************************/

static INT32 Update_errortype_counter(vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        trigger_1_check_point
 *
 *****************************************************************/

static INT32 Trigger_1_check_point(vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        trigger_2_check_point
 *
 *
 *****************************************************************/

static INT32 Trigger_2_check_point(vtask)
      VTASK *vtask;
{
      Analyze_digits(vtask);
}
/*****************************************************************
 *
 *    STATIC FUNCTION:        call_information_request
 *
 *****************************************************************/

static INT32 Call_Information_Request (vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        event_report_bcsm
 *
 *****************************************************************/

static INT32 Event_Report_BCSM (vtask)
      VTASK *vtask;
{ }
```

```
/*************************************************************
 *
 *    STATIC FUNCTION:        call_information_report
 *
 *************************************************************/

static INT32 Call_Information_Report (vtask)
      VTASK *vtask;
{ }
/*************************************************************
 *
 *    STATIC FUNCTION:        hold_call_in_network
 *
 *************************************************************/

static INT32 Hold_Call_In_Network (vtask)
      VTASK *vtask;
{ }
/*************************************************************
 *
 *    STATIC FUNCTION:        request_report_bcsm_Event
 *
 *************************************************************/

static INT32 Request_Report_BCSM_Event (vtask)
      VTASK *vtask;
{ }
/*************************************************************
 *
 *    STATIC FUNCTION:        reset_timer
 *
 *************************************************************/

static INT32 Reset_Timer (vtask)
      VTASK *vtask;
{ }
/*************************************************************
 *
 *    STATIC FUNCTION:        furnish_charging_information
 *
 *************************************************************/

static INT32 Furnish_Charging_Information (vtask)
      VTASK *vtask;
{ }
/*************************************************************
 *
 *    STATIC FUNCTION:        apply_charging
 *
 *************************************************************/

static INT32 Apply_Charging (vtask)
      VTASK *vtask;
{ }
/*************************************************************
 *
 *    STATIC FUNCTION:        request_notification_charging_Event
 *
 *************************************************************/

static INT32 Request_Notification_Charging_Event (vtask)
      VTASK *vtask;
{ }
/*************************************************************
 *
 *    STATIC FUNCTION:        send_charging_information
 *
 *************************************************************/

static INT32 Send_Charging_Information (vtask)
      VTASK *vtask;
{ }
/*************************************************************
 *
 *    STATIC FUNCTION:        establish_temporary_connection
 *
 *************************************************************/

static INT32 Establish_Temporary_Connection (vtask)
      VTASK *vtask;
{ }
/*************************************************************
 *
 *    STATIC FUNCTION:        connect
 *
 *************************************************************/

static INT32 Connect_ (vtask)
      VTASK *vtask;
{ }
/*************************************************************
 *
 *    STATIC FUNCTION:        collect_information
 *
 *************************************************************/

static INT32 Collect_Information_ (vtask)
      VTASK *vtask;
{ }
/*************************************************************
 *
 *    STATIC FUNCTION:        connect_to_resource
 *
 *************************************************************/

static INT32 Connect_To_Resource (vtask)
      VTASK *vtask;
{ }
```

```
/*****************************************************************
 *
 *    STATIC FUNCTION:        analyze_information
 *
 *****************************************************************/

static INT32 Analyze_Information (vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        select_route
 *
 *****************************************************************/

static INT32 Select_Route (vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        select_facility
 *
 *****************************************************************/

static INT32 Select_Facility (vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        continue
 *
 *****************************************************************/

static INT32 Continue_ (vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        release_call
 *
 *****************************************************************/

static INT32 Release_Call (vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        call_information
 *
 *****************************************************************/

static INT32 Call_Information (vtask)
      VTASK *vtask;
{ }

/*****************************************************************
 *
 *    STATIC FUNCTION:        SspVtaskInit
 *
 *****************************************************************/

static void sspVtaskInit (vtask)
      VTASK *vtask;
{
      vtask -> ref = (BYTE_TYPE *) wzalloc(sizeof(SSP_REF));
}

/*****************************************************************
 *
 *    STATIC FUNCTION:        SspConnInit
 *
 *****************************************************************/

static void sspConnInit(vtask)
      VTASK *vtask;
{
      INT32 *var = vtask -> var;

      var[Dialogue_Id] = ((vtask -> ownCEPid) << 8) + 1;
      var[Active_Dialogue] = 1;
      var[Aonhook] = 0;
      var[Bonhook] = 0;
}

/*****************************************************************
 *
 *    STATIC FUNCTION:        Ssp_body
 *
 *****************************************************************/

void ssp_body(msg)
      MESSAGE *msg;
{
      TCS_PARAM_BLOCK *params;

      switch (msg -> primitive.selector)
            {
                  case DOWN:
                        params = (TCS_PARAM_BLOCK *) msg -> parameters;
                        msg -> dstCEPid = (params -> dialogue_id) >> 8;
                        executeConnection(msg);
                        break;

                  default:
                        executeConnection(msg);
                        break;
            }
}
```

```
/*****************************************************************
 *
 *    STATIC FUNCTION:        SspTypeInit
 *
 *****************************************************************/

static void sspTypeInit (type)
    TASK_TYPE *type;
{
    type -> initVtask                                   = sspVtaskInit;

    type -> body                                        = ssp_body;

    type -> connections.count                           = get_maxcon();
    type -> connections.initFunc                        = sspConnInit;
    type -> connections.initialState                    = Idle;

    type -> connections.prims[UP].connCreatePrim        = Check_Trigger;
    type -> connections.prims[UP].connFailPrim          = Ssp_Congestion;
    type -> connections.prims[DOWN].connCreatePrim      = Tc_Begin_Indication;
    type -> connections.prims[DOWN].connFailPrim        = tc_end_request;

    type -> params[UP].putParameters                    = putsspSrvParameters;
    type -> params[UP].getParameters                    = getsspSrvParameters;
    type -> params[UP].freeParameters                   = freesspSrvParameters;
    type -> params[DOWN].putParameters                  = puttcSrvParameters;
    type -> params[DOWN].getParameters                  = gettcSrvParameters;
    type -> params[PEER].putParameters                  = putsspPeerParameters;
    type -> params[RES1].putParameters                  = putsspstorParameters;
    type -> params[RES1].getParameters                  = getsspstorParameters;

    type -> proc[analyze_digits()]                      = Analyze_digits;
    type -> proc[normal_call_handling()]                = Normal_call_handling;
    type -> proc[update_abort_counter()]                = Update_abort_counter;
    type -> proc[trigger_1_check_point()]               = Trigger_1_check_point;
    type -> proc[trigger_2_check_point()]               = Trigger_2_check_point;
    type -> proc[event_report_bcsm()]                   = Event_Report_BCSM;
    type -> proc[hold_call_in_network()]                = Hold_Call_In_Network;
    type -> proc[reset_timer()]                         = Reset_Timer;
    type -> proc[apply_charging()]                      = Apply_Charging;
    type -> proc[connect()]                             = Connect_;
    type -> proc[collect_information()]                 = Collect_Information_;
    type -> proc[connect_to_resource()]                 = Connect_To_Resource;
    type -> proc[analyze_information()]                 = Analyze_Information;
    type -> proc[select_route()]                        = Select_Route;
    type -> proc[select_facility()]                     = Select_Facility;
    type -> proc[continue()]                            = Continue_;
    type -> proc[release_call()]                        = Release_Call;
    type -> proc[call_information()]                    = Call_Information;
    type -> proc[send_charging_information()]           = Send_Charging_Information;
    type -> proc[request_report_bcsm_Event()]           = Request_Report_BCSM_Event;
    type -> proc[call_information_request()]            = Call_Information_Request;
    type -> proc[call_information_report()]             = Call_Information_Report;
    type -> proc[update_errortype_counter()]            = Update_errortype_counter;
    type -> proc[create_call_detail_record()]           = Create_call_detail_record;
    type -> proc[play_announcement_to_caller()]         = Play_announcement_to_caller;
    type -> proc[furnish_charging_information()]        = Furnish_Charging_Information;
    type -> proc[establish_temporary_connection()]      = Establish_Temporary_Connection;
    type -> proc[request_notification_charging_Event()] = Request_Notification_Charging_Event;
}

/*****************************************************************
 *
 *    GLOBAL FUNCTION:        sspAssoc
 *
 *****************************************************************/

void sspAssoc(typeInitPtr, fileList)
    void (**typeInitPtr) ();
    char *fileList [];
{
    fileList [PEER]     = "ssp.in";
    fileList [UP]       = "ssp.if";
    fileList [RES1]     = "sspstor.if";
    fileList [DOWN]     = "tc.if";
    fileList [AUTO]     = "ssp.aut";

    *typeInitPtr        = sspTypeInit;
}
```

D.2 SspEncod.c

```
/*****************************************************************
 *
 *   DATE:              31.08.1994
 *
 *   AUTHORS:           Ahti Rissanen
 *
 *   MODULE NAME:       SSPENCOD.C
 *
 *****************************************************************/

/*****************************************************************
 *
 *   FUNCTION:      Checkeventmask
 *
 *****************************************************************/

static void checkeventmask(ev, var)
    EVENT_MASK *ev;
    INT32 *var;
{
    EVENTBITS *req = &(ev -> requested_Event);
    EVENTBITS *rep = &(ev -> reporting_method);

    if(var[Active_Dialogue] > 1 || var[Dialog_End] == 1)
        {
            if(var[Dialog_End] == 1)
                {
                    var[Dialog_End] = 0;
                     --var[Active_Dialogue];
                     --var[Dialogue_Id];
                    var[Too_Tired] = 0;
                }
            return;
        }

    var[Event_Mask]             = 0;
    var[Response_Required]      = 0;
    var[Event_Parameter]        = 0;

    if(req -> call_delivered > 0 || req -> answered > 0 || req -> cleared > 0 ||
        req -> feature_requested > 0 || req -> disconnected > 0 ||
        req -> release_timeout > 0 || req -> invalid_information > 0 ||
        req -> network_busy > 0 || req -> busy_reported > 0 ||
        req -> resource_clear > 0 || req -> timeout_occurred > 0)
        {
            var[Event_Mask] = 1;
            var[Event_Parameter] = 1;
        }

    if(rep -> call_delivered > 0 || rep -> answered > 0 || rep -> cleared > 0 ||
        rep -> feature_requested > 0 || rep -> disconnected > 0 ||
        rep -> release_timeout > 0 || rep -> invalid_information > 0 ||
        rep -> network_busy > 0 || rep -> busy_reported > 0 ||
        rep -> resource_clear > 0 || rep -> timeout_occurred > 0)
        {
            var[Event_Mask] = 1;
            var[Response_Required] = 1;
        }

    var[Call_Delivered_Event]       = req -> call_delivered * 10 + rep -> call_delivered;
    var[Answered_Event]             = req -> answered * 10 + rep -> answered;
    var[Cleared_Event]              = req -> cleared * 10 + rep -> cleared;
    var[Feature_Requested_Event]    = req -> feature_requested * 10 + rep -> feature_requested;
    var[Disconnected_Event]         = req -> disconnected * 10 + rep -> disconnected;
    var[Release_Timeout_Event]      = req -> release_timeout * 10 + rep -> release_timeout;
    var[Invalid_Info_Event]         = req -> invalid_information * 10 + rep -> invalid_information;
    var[Network_Busy_Event]         = req -> network_busy * 10 + rep -> network_busy;
    var[Busy_Reported_Event]        = req -> busy_reported * 10 + rep -> busy_reported;
    var[Resource_Clear_Event]       = req -> resource_clear * 10 + rep -> resource_clear;
    var[Timeout_Occurred_Event]     = req -> timeout_occurred * 10 + rep -> timeout_occurred;
}

/*****************************************************************
 *
 *   FUNCTION:      Get_fields
 *
 *****************************************************************/

static void get_fields(var, ref)
    INT32 *var;
    SSP_REF *ref;
{
    int i;

    for(i = 0; i < 4;i++)
        {
            ref -> storblock[i].ann             = ref -> peerparameters.storblock[i].ann;
            ref -> storblock[i].interruptable   = ref -> peerparameters.storblock[i].interruptable;
            ref -> storblock[i].digits          = ref -> peerparameters.storblock[i].digits;
            ref -> storblock[i].len             = ref -> peerparameters.storblock[i].len;
        }
}

/*****************************************************************
 *
 *   FUNCTION:      SspEncode
 *
 *   PARAMETERS:
 *
 *****************************************************************/

static void     sspEncode(pdu, vtask)
    uint16      pdu;
    VTASK *vtask;
{
    SSP_REF *ref = (SSP_REF *) vtask -> ref;
    INT32 *var = vtask -> var;
    int i;
```

```
switch (pdu)
    {
        case AnalyzedInformation:
            ref -> peerparameters.communication_type = (INT16) var[Communication_Type];
            strcpy(ref -> peerparameters.collected_info, ref -> datadigits);
            strcpy(ref -> peerparameters.CallingPartyID, ref -> u_cli);
            break;

        case OCalledPartyBusy:
            strcpy(ref -> peerparameters.collected_info, ref -> u_digits);
            break;

        case ApplyChargingReport:
            break;

        case AssistRequestInstructions:
            break;

        case CallInformationReport:
            break;

        case CollectedInformation:
            break;

        case EventNotificationCharging:
            break;

        case EventReportBCSM:
            break;

        case InitialDP:
            break;

        case OAnswer:
            break;

        case ODisconnect:
            break;

        case OMidCall:
            break;

        case ONoAnswer:
            break;

        case OriginationAttemptAuthorized:
            break;

        case RouteSelectFailure:
            strcpy(ref -> peerparameters.collected_info, ref -> u_digits);
            break;

        case ServiceFilteringResponse:
            break;

        case StatusReport:
            break;

        case TAnswer:
            break;

        case TCalledPartyBusy:
            break;

        case TDisconnect:
            break;

        case TermAttemptAuthorized:
            break;

        case TMidCall:
            break;

        case TNoAnswer:
            break;

        case SpecializedResourceReport:
            break;

        default:
            break;
    }
}
```

D.3 SspDecod.c

```
/*****************************************************************
 *
 *   DATE:              31.08.1994
 *
 *   AUTHORS:           Ahti Rissanen
 *
 *   MODULE NAME:       SSPDECOD.C
 *
 *****************************************************************/

static void     sspDecode(vtask)
    VTASK *vtask;
{
    SSP_REF *ref = (SSP_REF *) vtask -> ref;
    AMAPARAMETERS *amap = &(ref -> peerparameters.ama);
    sint32 *var = vtask -> var;
    EVENT_MASK *ev = &(ref -> peerparameters.event_mask);
    int i, dis = 0;

    switch((int) var[Operation_Code])
        {
            case ActivityTest:
                break;

            case AnalyzeInformation:
                checkeventmask(ev, var);
                strcpy(ref -> datadigits, ref -> peerparameters.collected_info);
                traceLine(ref -> datadigits);*/
                break;

            case ApplyCharging:
                break;

            case CallGap:
                break;

            case CallInformationReport:
                break;

            case CallInformationRequest:
                checkeventmask(ev, var);
                break;

            case Cancel:
                break;

            case CancelStatusReport:
                break;

            case CollectInformation:
                checkeventmask(ev, var);
                get_fields(var, ref);
                break;

            case Connect:
                break;

            case ConnectToResource:
                break;

            case Continue:
                checkeventmask(ev, var);
                strcpy(ref -> datadigits, "EMPTY");
                break;

            case DisconnectForwardConnection:
                checkeventmask(ev, var);
                get_fields(var, ref);

                if (ref -> peerparameters.leg_ID == 0)
                    {
                        for(i = 0; i < 4;i++)
                            {
                                if(ref -> storblock[i].ann! = 0 && dis == 0)
                                    {
                                        dis = 1;
                                        ref -> peerparameters.leg_ID = 0;
                                    }
                            }
                    }

                if(ref -> peerparameters.leg_ID == 0 && dis == 0)
                    var[Disc_Case] = (INT32) 1;

                if(ref -> peerparameters.leg_ID == 1)
                    var[Disc_Case] = (INT32) 2;
                break;

            case EstablishTemporaryConnection:
                break;

            case FurnishChargingInformation:
                break;

            case HoldCallInNetwork:
                break;

            case InitiateCallAttempt:
                break;

            case ReleaseCall:
                break;

            case RequestCurrentStatusReport:
                break;

            case RequestEveryStatusChangeReport:
                break;

            case RequestFirstStatusMatchReport:
                break;
```

```
            case RequestNotificationChargingEvent:
                break;

            case RequestReportBCSMEvent:
                break;

            case ResetTimer:
                break;

            case SelectFacility:
                break;

            case SelectRoute:
                break;

            case SendChargingInformation:
                break;

            case PlayAnnouncement:
                break;

            case PromptAndCollectUserInformation:
                break;

            default:
                break;
        }
}
```

E Esimerkkiajo

```
/* Number translation service: number 98001111 is changed to 90-123456. */
: set trace tester off
Current trace: ( m v p a f t s i3)
New trace:     ( )

: ma end

: phone Lift s

yourCEPid :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :    User
DESTINATION     :    phone
PRIMITIVE       :    UP.Lift
SRC CEPid       :    -
DEST CEPid      :    -
TIME            :    25.8.1994 13.38:53.00
: END OF PARS
**********************************************************************

VOPS TASK TRACING    :

TASK            :    phone
VARIABLES       :    stor_CEP      :    000000    fields      :    000000
SHARED VAR.     :
FRAMES          :
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :    User
DESTINATION     :    phone.con_.1
PRIMITIVE       :    UP.Lift
SRC CEPid       :    -
DEST CEPid      :    1
TIME            :    25.8.1994 13.38:53.00
: END OF PARS
**********************************************************************

VOPS TASK TRACING    :

TASK            :    phone.con_.1
VARIABLES       :    stor_CEP      :    000000    fields      :    000000
SHARED VAR.     :
FRAMES          :
TIMERS          :
STATE           :    Idle
INPUT           :    UP.Lift
ACTIONS         :    if ( 0 == prompt() ) 98001111
                     { DOWN.Phone_Digits_Collected
                     to(Ph_Off_Hook) }
**********************************************************************

VOPS TASK TRACING AFTER ACTIONS    :

TASK            :    phone.con_.1
VARIABLES       :    stor_CEP      :    000000    fields      :    000000
SHARED VAR.     :
FRAMES          :
TIMERS          :

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS    :

TASK            :    phone
VARIABLES       :    stor_CEP      :    000000    fields      :    000000
SHARED VAR.     :
FRAMES          :
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :    phone.con_.1
DESTINATION     :    ccafA
PRIMITIVE       :    UP.Phone_Digits_Collected
SRC CEPid       :    1
DEST CEPid      :    -
TIME            :    25.8.1994 13.39:22.00
: END OF PARS
**********************************************************************

VOPS TASK TRACING    :

TASK            :    ccafA
VARIABLES       :    empty         :    000000    resp        :    000000
                     conn          :    000000    Phone_CEP   :    000000
SHARED VAR.     :
FRAMES          :    sendframe     :    000000    recframe    :    000000
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :    phone.con_.1
DESTINATION     :    ccafA.con_.1
PRIMITIVE       :    UP.Phone_Digits_Collected
SRC CEPid       :    1
DEST CEPid      :    1
```

```
TIME            :   25.8.1994 13.39:22.00
: END OF PARS
**********************************************************************

VOPS TASK TRACING    :

TASK            :   ccafA.con_.1
VARIABLES       :   empty       :   000000   resp          :   000000
                    conn        :   000000   Phone_CEP     :   000000
SHARED VAR.     :
FRAMES          :   sendframe   :   000000   recframe      :   000000
TIMERS          :
STATE           :   O_NullAndOriginatingAttemptAuthorized
INPUT           :   UP.Phone_Digits_Collected
ACTIONS         :   { ( conn = 0 ) ( empty = 0 ) enter_cli() DOWN.Check_Trigger
                      to(Analyzing_Information) }
**********************************************************************

VOPS TASK TRACING AFTER ACTIONS    :

TASK            :   ccafA.con_.1
VARIABLES       :   empty       :   000000   resp          :   000000
                    conn        :   000000   Phone_CEP     :   000001
SHARED VAR.     :
FRAMES          :   sendframe   :   000000   recframe      :   000000
TIMERS          :
**********************************************************************

VOPS TASK TRACING AFTER ACTIONS    :

TASK            :   ccafA
VARIABLES       :   empty       :   000000   resp          :   000000
                    conn        :   000000   Phone_CEP     :   000000
SHARED VAR.     :
FRAMES          :   sendframe   :   000000   recframe      :   000000
TIMERS          :
**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   ccafA.con_.1
DESTINATION     :   ssp
PRIMITIVE       :   UP.Check_Trigger
SRC CEPid       :   1
DEST CEPid      :   -
TIME            :   25.8.1994 13.39:22.00
: END OF PARS
**********************************************************************

VOPS TASK TRACING    :

TASK            :   ssp
VARIABLES       :   AbandonDP           :   000000   Active_Dialogue      :   000000
                    Aonhook             :   000000   Bonhook              :   000000
                    Call_Gapping_Event  :   000000   Call_Information_Rep :   000000
                    Call_Information_Req:   000000   Communication_Type   :   000000
                    Dialogue_Id         :   000000   DisconnectDP         :   000000
                    EDP_Event           :   000000   Event_Mask           :   000000
                    Event_Parameter     :   000000   Service_Blocking_eve  :   000000
                    Answered_Event      :   000000   Bcsm_State           :   000000
                    Busy_Reported_Event :   000000   Call_Delivered_Event :   000000
                    Cleared_Event       :   000000   Dialog_End           :   000000
                    Disc_Case           :   000000   Disconnected_Event   :   000000
                    Error_Code          :   000000   feature_requested_ev  :   000000
                    First_Ann           :   000000   Invalid_Info_Event   :   000000
                    Max_Call_Time       :   000000   Network_Busy_Event   :   000000
                    Operation_Code      :   000000   Origin_On_Hook       :   000000
                    Phone_CEP           :   000000   Release_Timeout_even :   000000
                    Resource_Clear_Event:   000000   Response_Required    :   000000
                    Route_Failure_Event :   000000   Second_Ann           :   000000
                    Send_Answer         :   000000   Thru_Connected       :   000000
                    Timeout_Occurred_Eve:   000000   Too_Tired            :   000000
                    Trigger             :   000000
SHARED VAR.     :
FRAMES          :
TIMERS          :
**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   ccafA.con_.1
DESTINATION     :   ssp.con_.1
PRIMITIVE       :   UP.Check_Trigger
SRC CEPid       :   1
DEST CEPid      :   1
TIME            :   25.8.1994 13.39:23.00
: END OF PARS
**********************************************************************

VOPS TASK TRACING    :

TASK            :   ssp.con_.1
VARIABLES       :   AbandonDP           :   000000   Active_Dialogue      :   000001
                    Aonhook             :   000000   Bonhook              :   000000
                    Call_Gapping_Event  :   000000   Call_Information_Rep :   000000
                    Call_Information_Req:   000000   Communication_Type   :   000000
                    Dialogue_Id         :   000257   DisconnectDP         :   000000
                    EDP_Event           :   000000   Event_Mask           :   000000
                    Event_Parameter     :   000000   Service_Blocking_eve  :   000000
                    Answered_Event      :   000000   Bcsm_State           :   000000
                    Busy_Reported_Event :   000000   Call_Delivered_Event :   000000
                    Cleared_Event       :   000000   Dialog_End           :   000000
                    Disc_Case           :   000000   Disconnected_Event   :   000000
                    Error_Code          :   000000   feature_requested_ev  :   000000
                    First_Ann           :   000000   Invalid_Info_Event   :   000000
                    Max_Call_Time       :   000000   Network_Busy_Event   :   000000
                    Operation_Code      :   000000   Origin_On_Hook       :   000000
                    Phone_CEP           :   000000   Release_Timeout_even :   000000
                    Resource_Clear_Event:   000000   Response_Required    :   000000
```

```
                    Route_Failure_Event     :   000000   Second_Ann          :   000000
                    Send_Answer             :   000000   Thru_Connected      :   000000
                    Timeout_Occurred_Eve    :   000000   Too_Tired          :   000000
                    Trigger                 :   000000
SHARED VAR.         :
FRAMES              :
TIMERS              :
STATE               :   Idle
INPUT               :   UP.Check_Trigger
ACTIONS             :   if ( Call_Gapping_Event == 1 )
                        if ( Service_Blocking_Event == 1 )
                             { ( Error_Code = 0 )
                               ( Thru_Connected = 0 )
                               ( Send_Answer = 1 )
                               ( Origin_On_Hook = 0 ) }
                        { analyze_digits()
                          trigger_1_check_point()
                          if ( Trigger == 1 ) }
                             { ( Communication_Type = 2 )
                               PEER.AnalyzedInformation
                               start(WF_Instructions_Timer)
                               DOWN.Tc_Begin_Request
                               to(WF_Instructions) }

*********************************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK                :   ssp.con_.1
VARIABLES           :   AbandonDP               :   000000   Active_Dialogue      :   000001
                        Aonhook                 :   000000   Bonhook             :   000000
                        Call_Gapping_Event      :   000000   Call_Information_Rep  :   000000
                        Call_Information_Req    :   000000   Communication_Type   :   000002
                        Dialogue_Id             :   000257   DisconnectDP         :   000000
                        EDP_Event               :   000000   Event_Mask          :   000000
                        Event_Parameter         :   000000   Service_Blocking_eve :   000000
                        Answered_Event          :   000000   Bcsm_State          :   000000
                        Busy_Reported_Event     :   000000   Call_Delivered_Event :   000000
                        Cleared_Event           :   000000   Dialog_End          :   000000
                        Disc_Case               :   000000   Disconnected_Event  :   000000
                        Error_Code              :   000000   feature_requested_ev :   000000
                        First_Ann               :   000000   Invalid_Info_Event  :   000000
                        Max_Call_Time           :   000000   Network_Busy_Event  :   000000
                        Operation_Code          :   000002   Origin_On_Hook       :   000000
                        Phone_CEP               :   000001   Release_Timeout_even :   000000
                        Resource_Clear_Event    :   000000   Response_Required    :   000000
                        Route_Failure_Event     :   000000   Second_Ann          :   000000
                        Send_Answer             :   000001   Thru_Connected      :   000000
                        Timeout_Occurred_Eve    :   000000   Too_Tired          :   000000
                        Trigger                 :   000001
SHARED VAR.         :
FRAMES              :
TIMERS              :   WF_Instructions_time tick count = 60.00  time left = 59.00 s

*********************************************************************************

VOPS MESSAGE TRACING

SOURCE              :   ssp.con_.1
DESTINATION         :   tc
PRIMITIVE           :   UP.tc_invoke_request
SRC CEPid           :   1
DEST CEPid          :   -
TIME                :   25.8.1994 13.39:28.00
: END OF PARS

*********************************************************************************

VOPS TASK TRACING   :

TASK                :   tc
VARIABLES           :   target          :   000000   dialogid   :   000000
SHARED VAR.         :
FRAMES              :
TIMERS              :
STATE               :   connected
INPUT               :   UP.tc_invoke_request
ACTIONS             :   add_to_buffer()

*********************************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK                :   tc
VARIABLES           :   target          :   000000   dialogid   :   000257
SHARED VAR.         :
FRAMES              :
TIMERS              :

*********************************************************************************

VOPS MESSAGE TRACING

SOURCE              :   ssp.con_.1
DESTINATION         :   tc
PRIMITIVE           :   UP.Tc_Begin_Request
SRC CEPid           :   1
DEST CEPid          :   -
TIME                :   25.8.1994 13.39:28.00
: END OF PARS

*********************************************************************************

VOPS TASK TRACING   :

TASK                :   tc
VARIABLES           :   target          :   000000   dialogid   :   000257
SHARED VAR.         :
FRAMES              :
TIMERS              :
STATE               :   connected
INPUT               :   UP.Tc_Begin_Request
ACTIONS             :   { PEER.TC_BEGIN
                          send_buffer() }
```

```
**********************************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK            :   tc
VARIABLES       :   target          :   000000   dialogid  :   000257
SHARED VAR.     :
FRAMES          :
TIMERS          :

**********************************************************************************

VOPS MESSAGE TRACING

SOURCE          :   tc
DESTINATION     :   scl
PRIMITIVE       :   UP.n_uDtr
SRC CEPid       :   -
DEST CEPid      :   -
TIME            :   25.8.1994 13.39:29.00
PARAMETERS      :

 frame contains following data:
01 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 FF FF 02 00 FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 39 38 30 30 31 31 ??????????980011
31 31 00 42 45 52 20 49 4E 20 54 48 45 20 46 49 11?BER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 39 30 31 31 31 32 ??????????901112
32 32 00 42 45 52 20 49 4E 20 54 48 45 20 46 49 22?BER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00             ????????????
: END OF PARS

**********************************************************************************

VOPS TASK TRACING    :

TASK            :   scl
VARIABLES       :
SHARED VAR.     :
FRAMES          :   SendFrame       :   000000   RecFrame     :   000000
TIMERS          :
STATE           :   Idle
INPUT           :   UP.n_uDtr
ACTIONS         :   { DOWN.n_cr
                    DOWN.n_dtr
                    to(connected) }

**********************************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK            :   scl
VARIABLES       :
SHARED VAR.     :
FRAMES          :   SendFrame       :   000000   RecFrame     :   000000
TIMERS          :

**********************************************************************************

VOPS MESSAGE TRACING

SOURCE          :   tc
DESTINATION     :   scl
PRIMITIVE       :   UP.n_uDtr
SRC CEPid       :   -
DEST CEPid      :   -
TIME            :   25.8.1994 13.39:31.00
PARAMETERS      :

 frame contains following data:
05 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 ????????????????
00 00 00 00 00 00 FF FF 02 00 FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
```

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 39 38 30 30 31 31 ??????????980011
31 31 00 42 45 52 20 49 4E 20 54 48 45 20 46 49 11?BER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 39 30 31 31 31 32 ??????????901112
32 32 00 42 45 52 20 49 4E 20 54 48 45 20 46 49 22?BER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00                   ??????????
: END OF PARS

**********************************************************************

VOPS TASK TRACING    :

TASK          :   scl
VARIABLES     :
SHARED VAR.   :
FRAMES        :   SendFrame    :   000000   RecFrame    :   000000
TIMERS        :
STATE         :   connected
INPUT         :   UP.n_uDtr
ACTIONS       :   DOWN.n_dtr

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS    :

TASK          :   scl
VARIABLES     :
SHARED VAR.   :
FRAMES        :   SendFrame    :   000000   RecFrame    :   000000
TIMERS        :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE        :   scl
DESTINATION   :   sclient
PRIMITIVE     :   UP.n_cr
SRC CEPid     :   -
DEST CEPid    :   -
TIME          :   25.8.1994  13.39:32.00
PARAMETERS    :

calledAddress :   haukka
: END OF PARS

**********************************************************************

VOPS TASK TRACING    :

TASK          :   sclient
VARIABLES     :   DEVICE_ID    :   000000
SHARED VAR.   :
FRAMES        :   SendFrame    :   000000   RecFrame    :   000000
TIMERS        :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE        :   scl
DESTINATION   :   sclient.con_.1
PRIMITIVE     :   UP.n_cr
SRC CEPid     :   -
DEST CEPid    :   1
TIME          :   25.8.1994  13.39:32.00
PARAMETERS    :

calledAddress :   haukka
: END OF PARS

**********************************************************************

VOPS TASK TRACING    :

TASK          :   sclient.con_.1
VARIABLES     :   DEVICE_ID    :   000000
SHARED VAR.   :
FRAMES        :   SendFrame    :   000000   RecFrame    :   000000
TIMERS        :
STATE         :   Idle
INPUT         :   UP.n_cr
ACTIONS       :   if ( socket_active_open() )
                  { UP.n_cc
                    to(connected) }

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS    :

TASK          :   sclient.con_.1
VARIABLES     :   DEVICE_ID    :   000001
SHARED VAR.   :
FRAMES        :   SendFrame    :   000000   RecFrame    :   000000
TIMERS        :
```

```
**********************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK            :   sclient
VARIABLES       :   DEVICE_ID   :   000000
SHARED VAR.     :
FRAMES          :   SendFrame   :   000000   RecFrame   :   000000
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   scl
DESTINATION     :   sclient
PRIMITIVE       :   UP.n_dtr
SRC CEPid       :   -
DEST CEPid      :   -
TIME            :   25.8.1994 13.39:34.00
PARAMETERS      :

 frame contains following data:
01 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 FF FF 02 00 FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 39 38 30 30 31 31 ??????????980011
31 31 00 42 45 52 20 49 4E 20 54 48 45 20 46 49 11?BER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 39 30 31 31 31 32 ??????????901112
32 32 00 42 45 52 20 49 4E 20 54 48 45 20 46 49 22?BER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00          ??????????
: END OF PARS

**********************************************************************

VOPS TASK TRACING    :

TASK            :   sclient
VARIABLES       :   DEVICE_ID   :   000000
SHARED VAR.     :
FRAMES          :   SendFrame   :   000000   RecFrame   :   000000
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   scl
DESTINATION     :   sclient.con_.1
PRIMITIVE       :   UP.n_dtr
SRC CEPid       :   -
DEST CEPid      :   1
TIME            :   25.8.1994 13.39:35.00
PARAMETERS      :

 frame contains following data:
01 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 FF FF 02 00 FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 39 38 30 30 31 31 ??????????980011
31 31 00 42 45 52 20 49 4E 20 54 48 45 20 46 49 11?BER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
```

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 39 30 31 31 31 32 ??????????901112
32 32 00 42 45 52 20 49 4E 20 54 48 45 20 46 49 22?BER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00                   ??????????
: END OF PARS

**********************************************************************

VOPS TASK TRACING    :

TASK            :   sclient.con_.1
VARIABLES       :   DEVICE_ID    :   000001
SHARED VAR.     :
FRAMES          :   SendFrame    :   000000   RecFrame    :   000000
TIMERS          :
STATE           :   connected
INPUT           :   UP.n_dtr
ACTIONS         :   socket_tx()

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK            :   sclient.con_.1
VARIABLES       :   DEVICE_ID    :   000001
SHARED VAR.     :
FRAMES          :   SendFrame    :   000000   RecFrame    :   000000
TIMERS          :

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK            :   sclient
VARIABLES       :   DEVICE_ID    :   000000
SHARED VAR.     :
FRAMES          :   SendFrame    :   000000   RecFrame    :   000000
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   scl
DESTINATION     :   sclient
PRIMITIVE       :   UP.n_dtr
SRC CEPid       :   -
DEST CEPid      :   -
TIME            :   25.8.1994 13.39:37.00
PARAMETERS      :

 frame contains following data:
05 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 ????????????????
00 00 00 00 00 00 FF FF 02 00 FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 39 38 30 30 31 31 ??????????980011
31 31 00 42 45 52 20 49 4E 20 54 48 45 20 46 49 11?BER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 39 30 31 31 31 32 ??????????901112
32 32 00 42 45 52 20 49 4E 20 54 48 45 20 46 49 22?BER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00                   ??????????
: END OF PARS

**********************************************************************

VOPS TASK TRACING    :

TASK            :   sclient
VARIABLES       :   DEVICE_ID    :   000000
SHARED VAR.     :
FRAMES          :   SendFrame    :   000000   RecFrame    :   000000
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   scl
DESTINATION     :   sclient.con_.1
PRIMITIVE       :   UP.n_dtr
SRC CEPid       :   -
DEST CEPid      :   1
TIME            :   25.8.1994 13.39:39.00
```

```
PARAMETERS      :

 frame contains following data:
05 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 ????????????????
00 00 00 00 00 00 FF FF 02 00 FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 39 38 30 30 31 31 ??????????980011
31 31 00 42 45 52 20 49 4E 20 54 48 45 20 46 49 11?BER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 39 30 31 31 31 32 ??????????901112
32 32 00 42 45 52 20 49 4E 20 54 48 45 20 46 49 22?BER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00                   ??????????
: END OF PARS

*********************************************************************************

VOPS TASK TRACING     :

TASK            :   sclient.con_.1
VARIABLES       :   DEVICE_ID    :   000001
SHARED VAR.     :
FRAMES          :   SendFrame    :   000000   RecFrame    :   000000
TIMERS          :
STATE           :   connected
INPUT           :   UP.n_dtr
ACTIONS         :   socket_tx()

*********************************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   sclient.con_.1
VARIABLES       :   DEVICE_ID    :   000001
SHARED VAR.     :
FRAMES          :   SendFrame    :   000000   RecFrame    :   000000
TIMERS          :

*********************************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   sclient
VARIABLES       :   DEVICE_ID    :   000000
SHARED VAR.     :
FRAMES          :   SendFrame    :   000000   RecFrame    :   000000
TIMERS          :

*********************************************************************************

VOPS MESSAGE TRACING

SOURCE          :   sclient.con_.1
DESTINATION     :   scl
PRIMITIVE       :   DOWN.n_cc
SRC CEPid       :   1
DEST CEPid      :   -
TIME            :   25.8.1994  13.39:41.00
PARAMETERS      :
: END OF PARS

*********************************************************************************

VOPS TASK TRACING     :

TASK            :   scl
VARIABLES       :
SHARED VAR.     :
FRAMES          :   SendFrame    :   000000   RecFrame    :   000000
TIMERS          :
STATE           :   connected
INPUT           :   DOWN.n_cc
ACTIONS         :   (rule ANYSTATE <input> used)

*********************************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   scl
VARIABLES       :
SHARED VAR.     :
FRAMES          :   SendFrame    :   000000   RecFrame    :   000000
TIMERS          :
```

```
**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   sclient.con_.1
DESTINATION     :   sclient.con_.1
PRIMITIVE       :   DOWN.l_dti
SRC CEPid       :   1
DEST CEPid      :   1
TIME            :   25.8.1994 13.39:44.00
PARAMETERS      :

 frame contains following data:
03 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 FF FF FF FF FF FF FF FF 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 31 32 33 34 35 36 ??????????123456
00 55 4D 42 45 52 20 49 4E 20 54 48 45 20 46 49 ?UMBER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 20 ??????????EMPTY
4E 55 4D 42 45 52 20 49 4E 20 54 48 45 20 46 49 NUMBER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00          ??????????
: END OF PARS

**********************************************************************

VOPS TASK TRACING    :

TASK            :   sclient.con_.1
VARIABLES       :   DEVICE_ID    :   000001
SHARED VAR.     :
FRAMES          :   SendFrame    :   000000    RecFrame    :   000000
TIMERS          :
STATE           :   connected
INPUT           :   DOWN.l_dti
ACTIONS         :   UP.n_dti

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   sclient.con_.1
VARIABLES       :   DEVICE_ID    :   000001
SHARED VAR.     :
FRAMES          :   SendFrame    :   000000    RecFrame    :   000000
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   sclient.con_.1
DESTINATION     :   scl
PRIMITIVE       :   DOWN.n_dti
SRC CEPid       :   1
DEST CEPid      :   -
TIME            :   25.8.1994 13.39:45.00
PARAMETERS      :

 frame contains following data:
03 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 FF FF FF FF FF FF FF FF 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
```

```
00 00 00 00 00 00 00 00 00 00 31 32 33 34 35 36 ??????????123456
00 55 4D 42 45 52 20 49 4E 20 54 48 45 20 46 49 ?UMBER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 20 ??????????EMPTY
4E 55 4D 42 45 52 20 49 4E 20 54 48 45 20 46 49 NUMBER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00                   ??????????
: END OF PARS

**********************************************************************

VOPS TASK TRACING     :

TASK            :   scl
VARIABLES       :
SHARED VAR.     :
FRAMES          :   SendFrame     :   000000    RecFrame      :   000000
TIMERS          :
STATE           :   connected
INPUT           :   DOWN.n_dti
ACTIONS         :   UP.n_uDti

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   scl
VARIABLES       :
SHARED VAR.     :
FRAMES          :   SendFrame     :   000000    RecFrame      :   000000
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   scl
DESTINATION     :   tc
PRIMITIVE       :   DOWN.n_uDti
SRC CEPid       :   -
DEST CEPid      :   -
TIME            :   25.8.1994 13.39:47.00
PARAMETERS      :

 frame contains following data:
03 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 FF FF FF FF FF FF FF FF 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ????????????????
FF FF FF FF FF FF FF FF FF FF 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 31 32 33 34 35 36 ??????????123456
00 55 4D 42 45 52 20 49 4E 20 54 48 45 20 46 49 ?UMBER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 20 ??????????EMPTY
4E 55 4D 42 45 52 20 49 4E 20 54 48 45 20 46 49 NUMBER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ????????????????
00 00 00 00 00 00 00 00 00 00                   ??????????
: END OF PARS

**********************************************************************

VOPS TASK TRACING     :

TASK            :   tc
VARIABLES       :   target        :   000000    dialogid      :   000257
SHARED VAR.     :
FRAMES          :
TIMERS          :
STATE           :   connected
INPUT           :   PEER.TC_END
ACTIONS         :   UP.Tc_End_Indication

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   tc
VARIABLES       :   target        :   000000    dialogid      :   000257
SHARED VAR.     :
FRAMES          :
TIMERS          :
```

****

VOPS MESSAGE TRACING

SOURCE : tc
DESTINATION : ssp
PRIMITIVE : DOWN.Tc_End_Indication
SRC CEPid : -
DEST CEPid : 1
TIME : 25.8.1994 13.39:49.00
: END OF PARS

****

VOPS TASK TRACING :

TASK : ssp
VARIABLES :

AbandonDP	:	000000	Active_Dialogue	:	000000
Aonhook	:	000000	Bonhook	:	000000
Call_Gapping_Event	:	000000	Call_Information_Rep	:	000000
Call_Information_Req	:	000000	Communication_Type	:	000000
Dialogue_Id	:	000000	DisconnectDP	:	000000
EDP_Event	:	000000	Event_Mask	:	000000
Event_Parameter	:	000000	Service_Blocking_eve	:	000000
Answered_Event	:	000000	bcsm_STATE	:	000000
Busy_Reported_Event	:	000000	Call_Delivered_Event	:	000000
Cleared_Event	:	000000	Dialog_End	:	000000
Disc_Case	:	000000	Disconnected_Event	:	000000
Error_Code	:	000000	feature_requested_ev	:	000000
First_Ann	:	000000	Invalid_Info_Event	:	000000
Max_Call_Time	:	000000	Network_Busy_Event	:	000000
Operation_Code	:	000000	Origin_On_Hook	:	000000
Phone_CEP	:	000000	Release_Timeout_even	:	000000
Resource_Clear_Event	:	000000	Response_Required	:	000000
Route_Failure_Event	:	000000	Second_Ann	:	000000
Send_Answer	:	000000	Thru_Connected	:	000000
Timeout_Occurred_Eve	:	000000	Too_Tired	:	000000
Trigger	:	000000			

SHARED VAR. :
FRAMES :
TIMERS :

****

VOPS MESSAGE TRACING

SOURCE : tc
DESTINATION : ssp.con_.1
PRIMITIVE : DOWN.Tc_End_Indication
SRC CEPid : -
DEST CEPid : 1
TIME : 25.8.1994 13.39:49.00
: END OF PARS

****

VOPS TASK TRACING :

TASK : ssp.con_.1
VARIABLES :

AbandonDP	:	000000	Active_Dialogue	:	000001
Aonhook	:	000000	Bonhook	:	000000
Call_Gapping_Event	:	000000	Call_Information_Rep	:	000000
Call_Information_Req	:	000000	Communication_Type	:	000002
Dialogue_Id	:	000257	DisconnectDP	:	000000
EDP_Event	:	000000	Event_Mask	:	000000
Event_Parameter	:	000000	Service_Blocking_eve	:	000000
Answered_Event	:	000000	bcsm_STATE	:	000000
Busy_Reported_Event	:	000000	Call_Delivered_Event	:	000000
Cleared_Event	:	000000	Dialog_End	:	000000
Disc_Case	:	000000	Disconnected_Event	:	000000
Error_Code	:	000000	feature_requested_ev	:	000000
First_Ann	:	000000	Invalid_Info_Event	:	000000
Max_Call_Time	:	000000	Network_Busy_Event	:	000000
Operation_Code	:	000002	Origin_On_Hook	:	000000
Phone_CEP	:	000001	Release_Timeout_even	:	000000
Resource_Clear_Event	:	000000	Response_Required	:	000000
Route_Failure_Event	:	000000	Second_Ann	:	000000
Send_Answer	:	000001	Thru_Connected	:	000000
Timeout_Occurred_Eve	:	000000	Too_Tired	:	000000
Trigger	:	000001			

SHARED VAR. :
FRAMES :
TIMERS : WF_Instructions_time tick count = 60.00 time left = 37.00 s
STATE : WF_Instructions
INPUT : DOWN.Tc_End_Indication
ACTIONS :

****

VOPS TASK TRACING AFTER ACTIONS :

TASK : ssp.con_.1
VARIABLES :

AbandonDP	:	000000	Active_Dialogue	:	000001
Aonhook	:	000000	Bonhook	:	000000
Call_Gapping_Event	:	000000	Call_Information_Rep	:	000000
Call_Information_Req	:	000000	Communication_Type	:	000002
Dialogue_Id	:	000257	DisconnectDP	:	000000
EDP_Event	:	000000	Event_Mask	:	000000
Event_Parameter	:	000000	Service_Blocking_eve	:	000000
Answered_Event	:	000000	bcsm_STATE	:	000000
Busy_Reported_Event	:	000000	Call_Delivered_Event	:	000000
Cleared_Event	:	000000	Dialog_End	:	000000
Disc_Case	:	000000	Disconnected_Event	:	000000
Error_Code	:	000000	feature_requested_ev	:	000000
First_Ann	:	000000	Invalid_Info_Event	:	000000
Max_Call_Time	:	000000	Network_Busy_Event	:	000000
Operation_Code	:	000002	Origin_On_Hook	:	000000
Phone_CEP	:	000001	Release_Timeout_even	:	000000
Resource_Clear_Event	:	000000	Response_Required	:	000000
Route_Failure_Event	:	000000	Second_Ann	:	000000
Send_Answer	:	000001	Thru_Connected	:	000000
Timeout_Occurred_Eve	:	000000	Too_Tired	:	000000
Trigger	:	000001			

SHARED VAR. :
FRAMES :
TIMERS : WF_Instructions_time tick count = 60.00 time left = 36.00 s

```
**********************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK            :   ssp
VARIABLES       :   AbandonDP                 :   000000   Active_Dialogue          :   000000
                    Aonhook                   :   000000   Bonhook                  :   000000
                    Call_Gapping_Event        :   000000   Call_Information_Rep      :   000000
                    Call_Information_Req      :   000000   Communication_Type        :   000000
                    Dialogue_Id               :   000000   DisconnectDP             :   000000
                    EDP_Event                 :   000000   Event_Mask               :   000000
                    Event_Parameter           :   000000   Service_Blocking_eve     :   000000
                    Answered_Event            :   000000   bcsm_STATE               :   000000
                    Busy_Reported_Event       :   000000   Call_Delivered_Event     :   000000
                    Cleared_Event             :   000000   Dialog_End               :   000000
                    Disc_Case                 :   000000   Disconnected_Event       :   000000
                    Error_Code                :   000000   feature_requested_ev     :   000000
                    First_Ann                 :   000000   Invalid_Info_Event       :   000000
                    Max_Call_Time             :   000000   Network_Busy_Event       :   000000
                    Operation_Code            :   000000   Origin_On_Hook           :   000000
                    Phone_CEP                 :   000000   Release_Timeout_even     :   000000
                    Resource_Clear_Event      :   000000   Response_Required        :   000000
                    Route_Failure_Event       :   000000   Second_Ann               :   000000
                    Send_Answer               :   000000   Thru_Connected           :   000000
                    Timeout_Occurred_Eve      :   000000   Too_Tired                :   000000
                    Trigger                   :   000000
SHARED VAR.     :
FRAMES          :
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   sclient.con_.1
DESTINATION     :   sclient.con_.1
PRIMITIVE       :   DOWN.l_dti
SRC CEPid       :   1
DEST CEPid      :   1
TIME            :   25.8.1994  13.39:52.00
PARAMETERS      :

 frame contains following data:
06 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 FF FF FF FF FF FF FF FF 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF ??????????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ??????????????????
FF FF FF FF FF FF FF FF FF FF 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 31 32 33 34 35 36 ??????????123456
00 55 4D 42 45 52 20 49 4E 20 54 48 45 20 46 49 ?UMBER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 20 ??????????EMPTY
4E 55 4D 42 45 52 20 49 4E 20 54 48 45 20 46 49 NUMBER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????????
00 00 00 00 00 00 00 00 00 00                   ??????????
: END OF PARS

**********************************************************************

VOPS TASK TRACING    :

TASK            :   sclient.con_.1
VARIABLES       :   DEVICE_ID     :   000001
SHARED VAR.     :
FRAMES          :   SendFrame     :   000000   RecFrame      :   000000
TIMERS          :
STATE           :   connected
INPUT           :   DOWN.l_dti
ACTIONS         :   UP.n_dti

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK            :   sclient.con_.1
VARIABLES       :   DEVICE_ID     :.  000001
SHARED VAR.     :
FRAMES          :   SendFrame     ,:  000000   RecFrame      :   000000
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   sclient.con_.1
DESTINATION     :   scl
PRIMITIVE       :   DOWN.n_dti
SRC CEPid       :   1
DEST CEPid      :   -
TIME            :   25.8.1994  13.39:53.00
```

```
PARAMETERS      :

 frame contains following data:
06 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 ??????????????
00 00 00 00 00 00 FF FF FF FF FF FF FF FF 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 31 32 33 34 35 36 ??????????123456
00 55 4D 42 45 52 20 49 4E 20 54 48 45 20 46 49 ?UMBER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 20 ??????????EMPTY
4E 55 4D 42 45 52 20 49 4E 20 54 48 45 20 46 49 NUMBER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00                   ??????????
: END OF PARS

**********************************************************************************

VOPS TASK TRACING     :

TASK            :   scl
VARIABLES       :
SHARED VAR.     :
FRAMES          :   SendFrame     :   000000    RecFrame      :   000000
TIMERS          :
STATE           :   connected
INPUT           :   DOWN.n_dti
ACTIONS         :   UP.n_uDti

**********************************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   scl
VARIABLES       :
SHARED VAR.     :
FRAMES          :   SendFrame     :   000000    RecFrame      :   000000
TIMERS          :

**********************************************************************************

VOPS MESSAGE TRACING

SOURCE          :   scl
DESTINATION     :   tc
PRIMITIVE       :   DOWN.n_uDti
SRC CEPid       :   -
DEST CEPid      :   -
TIME            :   25.8.1994  13.39:55.00
PARAMETERS      :

 frame contains following data:
06 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 ??????????????
00 00 00 00 00 00 FF FF FF FF FF FF FF FF 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ??????????????
FF FF FF FF FF FF FF FF FF FF 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 00 ??????????EMPTY?
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 31 32 33 34 35 36 ??????????123456
00 55 4D 42 45 52 20 49 4E 20 54 48 45 20 46 49 ?UMBER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 45 4D 50 54 59 20 ??????????EMPTY
4E 55 4D 42 45 52 20 49 4E 20 54 48 45 20 46 49 NUMBER IN THE FI
45 4C 44 00 00 00 00 00 00 00 00 00 00 00 00 00 ELD?????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ??????????????
```

```
00 00 00 00 00 00 00 00 00 00                ??????????
: END OF PARS
*************************************************************************

VOPS TASK TRACING    :

TASK          :   tc
VARIABLES     :   target          :   000000   dialogid   :   000257
SHARED VAR.   :
FRAMES        :
TIMERS        :
STATE         :   connected
INPUT         :   PEER.TC_RESULT_L
ACTIONS       :   UP.tc_result_l_indication
*************************************************************************

VOPS TASK TRACING AFTER ACTIONS    :

TASK          :   tc
VARIABLES     :   target          :   000000   dialogid   :   000257
SHARED VAR.   :
FRAMES        :
TIMERS        :
*************************************************************************

VOPS MESSAGE TRACING

SOURCE        :   tc
DESTINATION   :   ssp
PRIMITIVE     :   DOWN.tc_result_l_indication
SRC CEPid     :   -
DEST CEPid    :   1
TIME          :   25.8.1994 13.39:56.00
: END OF PARS
*************************************************************************

VOPS TASK TRACING    :

TASK          :   ssp
VARIABLES     :   AbandonDP          :   000000   Active_Dialogue       :   000000
                  Aonhook            :   000000   Bonhook               :   000000
                  Call_Gapping_Event :   000000   Call_Information_Rep  :   000000
                  Call_Information_Req:  000000   Communication_Type    :   000000
                  Dialogue_Id        :   000000   DisconnectDP          :   000000
                  EDP_Event          :   000000   Event_Mask            :   000000
                  Event_Parameter    :   000000   Service_Blocking_eve   :   000000
                  Answered_Event     :   000000   bcsm_STATE            :   000000
                  Busy_Reported_Event:   000000   Call_Delivered_Event   :   000000
                  Cleared_Event      :   000000   Dialog_End            :   000000
                  Disc_Case          :   000000   Disconnected_Event     :   000000
                  Error_Code         :   000000   feature_requested_ev   :   000000
                  First_Ann          :   000000   Invalid_Info_Event     :   000000
                  Max_Call_Time      :   000000   Network_Busy_Event     :   000000
                  Operation_Code     :   000000   Origin_On_Hook         :   000000
                  Phone_CEP          :   000000   Release_Timeout_even   :   000000
                  Resource_Clear_Event:  000000   Response_Required      :   000000
                  Route_Failure_Event:   000000   Second_Ann            :   000000
                  Send_Answer        :   000000   Thru_Connected         :   000000
                  Timeout_Occurred_Eve:  000000   Too_Tired             :   000000
                  Trigger            :   000000
SHARED VAR.   :
FRAMES        :
TIMERS        :
*************************************************************************

VOPS MESSAGE TRACING

SOURCE        :   tc
DESTINATION   :   ssp.con_.1
PRIMITIVE     :   DOWN.tc_result_l_indication
SRC CEPid     :   -
DEST CEPid    :   1
TIME          :   25.8.1994 13.39:57.00
: END OF PARS
*************************************************************************

VOPS TASK TRACING    :

TASK          :   ssp.con_.1
VARIABLES     :   AbandonDP          :   000000   Active_Dialogue       :   000001
                  Aonhook            :   000000   Bonhook               :   000000
                  Call_Gapping_Event :   000000   Call_Information_Rep  :   000000
                  Call_Information_Req:  000000   Communication_Type    :   000002
                  Dialogue_Id        :   000257   DisconnectDP          :   000000
                  EDP_Event          :   000000   Event_Mask            :   000000
                  Event_Parameter    :   000000   Service_Blocking_eve   :   000000
                  Answered_Event     :   000000   bcsm_STATE            :   000000
                  Busy_Reported_Event:   000000   Call_Delivered_Event   :   000000
                  Cleared_Event      :   000000   Dialog_End            :   000000
                  Disc_Case          :   000000   Disconnected_Event     :   000000
                  Error_Code         :   000000   feature_requested_ev   :   000000
                  First_Ann          :   000000   Invalid_Info_Event     :   000000
                  Max_Call_Time      :   000000   Network_Busy_Event     :   000000
                  Operation_Code     :   000002   Origin_On_Hook         :   000000
                  Phone_CEP          :   000001   Release_Timeout_even   :   000000
                  Resource_Clear_Event:  000000   Response_Required      :   000000
                  Route_Failure_Event:   000000   Second_Ann            :   000000
                  Send_Answer        :   000001   Thru_Connected         :   000000
                  Timeout_Occurred_Eve:  000000   Too_Tired             :   000000
                  Trigger            :   000001
SHARED VAR.   :
FRAMES        :
TIMERS        :   WF_Instructions_time tick count = 60.00  time left = 29.00 s
STATE         :   WF_Instructions
INPUT         :   PEER.AnalyzeInformation
ACTIONS       :   ( stop (WF_Instructions_Timer)
                    if ( Aonhook > 0 ) )
                  ( stop (all)
                    ( Origin_On_Hook = 1 )
                    ( Aonhook = 0 )
```

```
if ( ( EDP_Event == 1 ) ||
     ( Call_Information_Report_Event == 1 ) ) )
          { ( Bcsm_State = 5 )
            UP.Ssp_ToAnalyzing }
{ analyze_digits()
trigger_1_check_point()
if ( Trigger == 1 ) }
{ UP.Ssp_Normal_Call
  normal_call_handling()
{ ( Call_Delivered_Event = 0 ) ( Answered_Event = 0 ) (
  Cleared_Event = 0 ) ( Feature_Requested_Event = 0 ) (
  Disconnected_Event = 0 ) ( Release_Timeout_Event = 0 ) (
  Invalid_Info_Event = 0 ) ( Network_Busy_Event = 0 ) (
  Busy_Reported_Event = 0 ) ( Resource_Clear_Event = 0 ) (
  Timeout_Occurred_Event = 0 ) ( Route_Failure_Event = 0 ) }
to(Idle) }
```

****

VOPS TASK TRACING AFTER ACTIONS :

TASK	:	ssp.con_.1					
VARIABLES	:	AbandonDP	:	000000	Active_Dialogue	:	000001
		Aonhook	:	000000	Bonhook	:	000000
		Call_Gapping_Event	:	000000	Call_Information_Rep	:	000000
		Call_Information_Req	:	000000	Communication_Type	:	000002
		Dialogue_Id	:	000257	DisconnectDP	:	000000
		EDP_Event	:	000000	Event_Mask	:	000000
		Event_Parameter	:	000000	Service_Blocking_eve	:	000000
		Answered_Event	:	000000	bcsm_STATE	:	000005
		Busy_Reported_Event	:	000000	Call_Delivered_Event	:	000000
		Cleared_Event	:	000000	Dialog_End	:	000000
		Disc_Case	:	000000	Disconnected_Event	:	000000
		Error_Code	:	000000	feature_requested_ev	:	000000
		First_Ann	:	000000	Invalid_Info_Event	:	000000
		Max_Call_Time	:	000000	Network_Busy_Event	:	000000
		Operation_Code	:	000001	Origin_On_Hook	:	000001
		Phone_CEP	:	000001	Release_Timeout_even	:	000000
		Resource_Clear_Event	:	000000	Response_Required	:	000000
		Route_Failure_Event	:	000000	Second_Ann	:	000000
		Send_Answer	:	000001	Thru_Connected	:	000000
		Timeout_Occurred_Eve	:	000000	Too_Tired	:	000000
		Trigger	:	000000			
SHARED VAR.	:						
FRAMES	:						
TIMERS	:						

****

VOPS TASK TRACING AFTER ACTIONS :

TASK	:	ssp					
VARIABLES	:	AbandonDP	:	000000	Active_Dialogue	:	000000
		Aonhook	:	000000	Bonhook	:	000000
		Call_Gapping_Event	:	000000	Call_Information_Rep	:	000000
		Call_Information_Req	:	000000	Communication_Type	:	000000
		Dialogue_Id	:	000000	DisconnectDP	:	000000
		EDP_Event	:	000000	Event_Mask	:	000000
		Event_Parameter	:	000000	Service_Blocking_eve	:	000000
		Answered_Event	:	000000	bcsm_STATE	:	000000
		Busy_Reported_Event	:	000000	Call_Delivered_Event	:	000000
		Cleared_Event	:	000000	Dialog_End	:	000000
		Disc_Case	:	000000	Disconnected_Event	:	000000
		Error_Code	:	000000	feature_requested_ev	:	000000
		First_Ann	:	000000	Invalid_Info_Event	:	000000
		Max_Call_Time	:	000000	Network_Busy_Event	:	000000
		Operation_Code	:	000000	Origin_On_Hook	:	000000
		Phone_CEP	:	000000	Release_Timeout_even	:	000000
		Resource_Clear_Event	:	000000	Response_Required	:	000000
		Route_Failure_Event	:	000000	Second_Ann	:	000000
		Send_Answer	:	000000	Thru_Connected	:	000000
		Timeout_Occurred_Eve	:	000000	Too_Tired	:	000000
		Trigger	:	000000			
SHARED VAR.	:						
FRAMES	:						
TIMERS	:						

****

VOPS MESSAGE TRACING

SOURCE	:	ssp.con_.1
DESTINATION	:	ccafA
PRIMITIVE	:	DOWN.Ssp_ToAnalyzing
SRC CEPid	:	1
DEST CEPid	:	1
TIME	:	25.8.1994 13.40:01.00
: END OF PARS		

****

VOPS TASK TRACING :

TASK	:	ccafA					
VARIABLES	:	empty	:	000000	resp	:	000000
		conn	:	000000	Phone_CEP	:	000000
SHARED VAR.	:						
FRAMES	:	sendframe	:	000000	recframe	:	000000
TIMERS	:						

****

VOPS MESSAGE TRACING

SOURCE	:	ssp.con_.1
DESTINATION	:	ccafA.con_.1
PRIMITIVE	:	DOWN.Ssp_ToAnalyzing
SRC CEPid	:	1
DEST CEPid	:	1
TIME	:	25.8.1994 13.40:03.00
: END OF PARS		

```
**********************************************************************

VOPS TASK TRACING    :

TASK          :  ccafA.con_.1
VARIABLES     :  empty      :  000000   resp       :  000000
                 conn       :  000000   Phone_CEP  :  000001
SHARED VAR.   :
FRAMES        :  sendframe  :  000000   recframe   :  000000
TIMERS        :
STATE         :  Analyzing_Information
INPUT         :  DOWN.Ssp_ToAnalyzing
ACTIONS       :  (rule ANYSTATE <input> used)
                   to(Analyzing_Information)

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK          :  ccafA.con_.1
VARIABLES     :  empty      :  000000   resp       :  000000
                 conn       :  000000   Phone_CEP  :  000001
SHARED VAR.   :
FRAMES        :  sendframe  :  000000   recframe   :  000000
TIMERS        :

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK          :  ccafA
VARIABLES     :  empty      :  000000   resp       :  000000
                 conn       :  000000   Phone_CEP  :  000000
SHARED VAR.   :
FRAMES        :  sendframe  :  000000   recframe   :  000000
TIMERS        :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE        :  ssp.con_.1
DESTINATION   :  ccafA
PRIMITIVE     :  DOWN.Ssp_Normal_Call
SRC CEPid     :  1
DEST CEPid    :  1
TIME          :  25.8.1994 13.40:03.00
: END OF PARS

**********************************************************************

VOPS TASK TRACING    :

TASK          :  ccafA
VARIABLES     :  empty      :  000000   resp       :  000000
                 conn       :  000000   Phone_CEP  :  000000
SHARED VAR.   :
FRAMES        :  sendframe  :  000000   recframe   :  000000
TIMERS        :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE        :  ssp.con_.1
DESTINATION   :  ccafA.con_.1
PRIMITIVE     :  DOWN.Ssp_Normal_Call
SRC CEPid     :  1
DEST CEPid    :  1
TIME          :  25.8.1994 13.40:04.00
: END OF PARS

**********************************************************************

VOPS TASK TRACING    :

TASK          :  ccafA.con_.1
VARIABLES     :  empty      :  000000   resp       :  000000
                 conn       :  000000   Phone_CEP  :  000001
SHARED VAR.   :
FRAMES        :  sendframe  :  000000   recframe   :  000000
TIMERS        :

Called number: 123456
STATE         :  Analyzing_Information
INPUT         :  DOWN.Ssp_Normal_Call
ACTIONS       :  if ( empty == 0 )
                 { ( conn = 1 )
                   PEER.ADDRESS
                   start (WF_Address_Complete_Tim)
                   to(Routing_AndAlerting) }

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK          :  ccafA.con_.1
VARIABLES     :  empty      :  000000   resp       :  000000
                 conn       :  000001   Phone_CEP  :  000001
SHARED VAR.   :
FRAMES        :  sendframe  :  000000   recframe   :  000000
TIMERS        :  wf_address_complete_ tick count = 10.00  time left = 10.00 s

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK          :  ccafA
VARIABLES     :  empty      :  000000   resp       :  000000
                 conn       :  000000   Phone_CEP  :  000000
SHARED VAR.   :
FRAMES        :  sendframe  :  000000   recframe   :  000000
TIMERS        :
```

```
**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   ccafA.con_.1
DESTINATION     :   ccafB
PRIMITIVE       :   PEER.ADDRESS
SRC CEPid       :   1
DEST CEPid      :   -
TIME            :   25.8.1994 13.40:05.00
: END OF PARS

**********************************************************************

VOPS TASK TRACING   :

TASK            :   ccafB
VARIABLES       :   empty       :   000000   resp        :   000000
                    conn        :   000000   Phone_CEP   :   000000
SHARED VAR.     :
FRAMES          :   sendframe   :   000000   recframe    :   000000
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   ccafA.con_.1
DESTINATION     :   ccafB.con_.1
PRIMITIVE       :   PEER.ADDRESS
SRC CEPid       :   1
DEST CEPid      :   1
TIME            :   25.8.1994 13.40:05.00
: END OF PARS

**********************************************************************

VOPS TASK TRACING   :

TASK            :   ccafB.con_.1
VARIABLES       :   empty       :   000000   resp        :   000000
                    conn        :   000000   Phone_CEP   :   000000
SHARED VAR.     :
FRAMES          :   sendframe   :   000000   recframe    :   000000
TIMERS          :
STATE           :   T_NullAndAuthorizeTerminationAttempt
INPUT           :   PEER.ADDRESS
ACTIONS         :   if ( 1 == testbusy() )
                    { PEER.ADDRESS_COMPLETE
                      delete ( recframe )
                      to(T_Alerting) }

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   ccafB.con_.1
VARIABLES       :   empty       :   000000   resp        :   000000
                    conn        :   000000   Phone_CEP   :   000000
SHARED VAR.     :
FRAMES          :   sendframe   :   000000   recframe    :   000000
TIMERS          :

**********************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   ccafB
VARIABLES       :   empty       :   000000   resp        :   000000
                    conn        :   000000   Phone_CEP   :   000000
SHARED VAR.     :
FRAMES          :   sendframe   :   000000   recframe    :   000000
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   ccafB.con_.1
DESTINATION     :   ccafA
PRIMITIVE       :   PEER.ADDRESS_COMPLETE
SRC CEPid       :   1
DEST CEPid      :   1
TIME            :   25.8.1994 13.40:06.00
: END OF PARS

**********************************************************************

VOPS TASK TRACING   :

TASK            :   ccafA
VARIABLES       :   empty       :   000000   resp        :   000000
                    conn        :   000000   Phone_CEP   :   000000
SHARED VAR.     :
FRAMES          :   sendframe   :   000000   recframe    :   000000
TIMERS          :

**********************************************************************

VOPS MESSAGE TRACING

SOURCE          :   ccafB.con_.1
DESTINATION     :   ccafA.con_.1
PRIMITIVE       :   PEER.ADDRESS_COMPLETE
SRC CEPid       :   1
DEST CEPid      :   1
TIME            :   25.8.1994 13.40:06.00
: END OF PARS

**********************************************************************

VOPS TASK TRACING   :

TASK            :   ccafA.con_.1
VARIABLES       :   empty       :   000000   resp        :   000000
                    conn        :   000001   Phone_CEP   :   000001
```

```
SHARED VAR.     :
FRAMES          :   sendframe       :   000000    recframe        :   000000
TIMERS          :   wf_address_complete_ tick count = 10.00  time left = 7.00 s
STATE           :   Routing_AndAlerting
INPUT           :   PEER.ADDRESS_COMPLETE
ACTIONS         :   (rule GLOBAL OTHER used)

*******************************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK            :   ccafA.con_.1
VARIABLES       :   empty       :   000000    resp            :   000000
                    conn        :   000001    Phone_CEP       :   000001
SHARED VAR.     :
FRAMES          :   sendframe       :   000000    recframe        :   000000
TIMERS          :   wf_address_complete_ tick count = 10.00  time left = 7.00 s

*******************************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK            :   ccafA
VARIABLES       :   empty       :   000000    resp            73
:   000000
                    conn        :   000000    Phone_CEP       :   000000
SHARED VAR.     :
FRAMES          :   sendframe       :   000000    recframe        :   000000
TIMERS          :

: sh sys

CVOPS running following tasks  :

        phone.con_.1    Ph_Off_Hook
                ssp     Idle
              ccafA     O_NullAndOriginatingAttemptAuthorized
                scl     connected
              ccafB     T_NullAndOriginatingAttemptAuthorized
             tester     Idle
              phone     Idle
            storserv    Idle
             output
                 tc     connected
        sclient.con_.1  connected
          ccafB.con_.1  T_Alerting
          ccafA.con_.1  Routing_AndAlerting
            sclient     Idle

: ccafb.con_.1 answer s

*******************************************************************************

VOPS MESSAGE TRACING

SOURCE          :   User
DESTINATION     :   ccafB.con_.1
PRIMITIVE       :   UP.answer
SRC CEPid       :   -
DEST CEPid      :   -
TIME            :   25.8.1994  13.40:23.00
: END OF PARS

*******************************************************************************

VOPS TASK TRACING  :

TASK            :   ccafB.con_.1
VARIABLES       :   empty       :   000000    resp            :   000000
                    conn        :   000000    Phone_CEP       :   000000
SHARED VAR.     :
FRAMES          :   sendframe       :   000000    recframe        :   000000
TIMERS          :
STATE           :   T_Alerting
INPUT           :   UP.answer
ACTIONS         :   { PEER.TAnswer
                      ( resp = 1 )
                      to(T_Active) }

*******************************************************************************

VOPS TASK TRACING AFTER ACTIONS  :

TASK            :   ccafB.con_.1
VARIABLES       :   empty       :   000000    resp            :   000001
                    conn        :   000000    Phone_CEP       :   000000
SHARED VAR.     :
FRAMES          :   sendframe       :   000000    recframe        :   000000
TIMERS          :

*******************************************************************************

VOPS MESSAGE TRACING

SOURCE          :   ccafB.con_.1
DESTINATION     :   ccafA
PRIMITIVE       :   PEER.TAnswer
SRC CEPid       :   1
DEST CEPid      :   1
TIME            :   25.8.1994  13.40:23.00
: END OF PARS

*******************************************************************************

VOPS TASK TRACING  :

TASK            :   ccafA
VARIABLES       :   empty       :   000000    resp            :   000000
                    conn        :   000000    Phone_CEP       :   000000
SHARED VAR.     :
FRAMES          :   sendframe       :   000000    recframe        :   000000
TIMERS          :

*******************************************************************************

VOPS MESSAGE TRACING
```

```
SOURCE          :   ccafB.con_.1
DESTINATION     :   ccafA.con_.1
PRIMITIVE       :   PEER.TAnswer
SRC CEPid       :   1
DEST CEPid      :   1
TIME            :   25.8.1994 13.40:24.00
: END OF PARS
*********************************************************************************

VOPS TASK TRACING    :

TASK            :   ccafA.con_.1
VARIABLES       :   empty       :   000000   resp        :   000000
                    conn        :   000001   Phone_CEP    :   000001
SHARED VAR.     :
FRAMES          :   sendframe   :   000000   recframe     :   000000
TIMERS          :
STATE           :   Routing_AndAlerting
INPUT           :   PEER.TAnswer
ACTIONS         :   to(O_Active)
*********************************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   ccafA.con_.1
VARIABLES       :   empty       :   000000   resp        :   000000
                    conn        :   000001   Phone_CEP    :   000001
SHARED VAR.     :
FRAMES          :   sendframe   :   000000   recframe     :   000000
TIMERS          :

: sh sys

CVOPS running following tasks   :

        phone.con_.1    Ph_Off_Hook
                ssp     Idle
              ccafA     O_NullAndOriginatingAttemptAuthorized
                scl     connected
              ccafB     T_NullAndOriginatingAttemptAuthorized
             tester     Idle
              phone     Idle
           storserv     Idle
             output
                 tc     connected
    sclient.con_.1      connected
      ccafB.con_.1      T_Active
      ccafA.con_.1      O_Active
            sclient     Idle

: r

: ccafb.con_.1 On_Hook s
*********************************************************************************

VOPS MESSAGE TRACING

SOURCE          :   User
DESTINATION     :   ccafB.con_.1
PRIMITIVE       :   UP.On_Hook
SRC CEPid       :   -
DEST CEPid      :   -
TIME            :   25.8.1994 13.41:09.00
: END OF PARS
*********************************************************************************

VOPS TASK TRACING    :

TASK            :   ccafB.con_.1
VARIABLES       :   empty       :   000000   resp        :   000001
                    conn        :   000000   Phone_CEP    :   000000
SHARED VAR.     :
FRAMES          :   sendframe   :   000000   recframe     :   000000
TIMERS          :
STATE           :   T_Active
INPUT           :   UP.On_Hook
ACTIONS         :   if ( resp == 1 )
                    { PEER.CLEAR_BACK
                      to(T_NullAndAuthorizeTerminationAttempt) }
*********************************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   ccafB.con_.1
VARIABLES       :   empty       :   000000   resp        :   000001
                    conn        :   000000   Phone_CEP    :   000000
SHARED VAR.     :
FRAMES          :   sendframe   :   000000   recframe     :   000000
TIMERS          :
*********************************************************************************

VOPS MESSAGE TRACING

SOURCE          :   ccafB.con_.1
DESTINATION     :   ccafA
PRIMITIVE       :   PEER.CLEAR_BACK
SRC CEPid       :   1
DEST CEPid      :   1
TIME            :   25.8.1994 13.41:09.00
: END OF PARS
*********************************************************************************

VOPS TASK TRACING    :

TASK            :   ccafA
VARIABLES       :   empty       :   000000   resp        :   000000
                    conn        :   000000   Phone_CEP    :   000000
SHARED VAR.     :
```

```
TIMERS          :
*****************************************************************************

VOPS MESSAGE TRACING

SOURCE          :   ccafB.con_.1
DESTINATION     :   ccafA.con_.1
PRIMITIVE       :   PEER.CLEAR_BACK
SRC CEPid       :   1
DEST CEPid      :   1
TIME            :   25.8.1994 13.41:09.00
: END OF PARS
*****************************************************************************

VOPS TASK TRACING    :

TASK            :   ccafA.con_.1
VARIABLES       :   empty      :   000000   resp        :   000000
                    conn       :   000001   Phone_CEP   :   000001
SHARED VAR.     :
FRAMES          :   sendframe  :   000000   recframe    :   000000
TIMERS          :
STATE           :   O_Active
INPUT           :   PEER.CLEAR_BACK
ACTIONS         :   to(O_NullAndOriginatingAttemptAuthorized)
*****************************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   ccafA.con_.1
VARIABLES       :   empty      :   000000   resp        :   000000
                    conn       :   000001   Phone_CEP   :   000001
SHARED VAR.     :
FRAMES          :   sendframe  :   000000   recframe    :   000000
TIMERS          :
*****************************************************************************

VOPS TASK TRACING AFTER ACTIONS   :

TASK            :   ccafA
VARIABLES       :   empty      :   000000   resp        :   000000
                    conn       :   000000   Phone_CEP   :   000000
SHARED VAR.     :
FRAMES          :   sendframe  :   000000   recframe    :   000000
TIMERS          :

: sh sys

CVOPS running following tasks   :

        phone.con_.1    Ph_Off_Hook
                ssp     Idle
              ccafA     O_NullAndOriginatingAttemptAuthorized
                scl     connected
              ccafB     T_NullAndOriginatingAttemptAuthorized
             tester     Idle
              phone     Idle
           storserv     Idle
             output
                 tc     connected
    sclient.con_.1      connected
            sclient     Idle

: E
```